NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

WEB-CENTRIC SYSTEMS IN SUPPORT OF ARGUMENTATION, NEGOTIATION, AND ORGANIZATIONAL MEMORY

by

Randal R. Vickers
and
Carl M. Wright

December 1997

Principal Advisor:	Tung X. Bui
Associate Advisor:	Suresh Sridhar

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 1997	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE WEB-CENTRIC SYSTEMS IN SUPPORT OF ARGUMENTATION, NEGOTIATION, AND ORGANIZATIONAL MEMORY

5. FUNDING NUMBERS

6. AUTHOR(S) Randal R. Vickers and Carl M. Wright

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITOR ING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The purpose of this thesis is to propose and demonstrate a new negotiation and argumentation medium. This medium will take advantage of the latest in web technologies while conducting a detailed analysis and design of a prototype web based decision support system to support on-line argumentation, claims, and team decisions. The information obtained from the application will be stored in an ODBC database, to be used as part of the organizational memory. Organization memory will significantly enhance an organizations ability to utilize historical data in conjunction with current decision making requirements. The findings in this study strongly support the strengths of the action-resource based argumentation system (ARBAS) model and indicate that future research and application development would significantly advance the fields of web-based negotiation and argumentation. A web-centric prototype developed during this research can be viewed at [http://www.cimnet.nps.navy.mil/thesis].

14. SUBJECT TERMS ARBAS, Argumentation, Browser, Cold Fusion, Corporate Knowledge, Database, Decision Making, Intranet, Negotiation, Organizational Memory, Web Browser, Web-centric

15. NUMBER OF PAGES
81

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

WEB-CENTRIC SYSTEMS IN SUPPORT OF AURGUMENTATION, NEGOTIATION, AND ORGANIZATIONAL MEMORY

Randal R. Vickers
Captain, United States Army
B.S., Texas A&M University, 1986

Carl M. Wright
Captain, United States Marine Corps
B.S., Augsburg College, 1990

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
December 1997

Authors: ______________________________
Randal R. Vickers

____
Carl M. Wright

Approved by: TBui
____
Tung X. Bui, Thesis Principal Advisor

SSridhar
____
Suresh Sridhar, Thesis Associate Advisor

Reuben T Harris
____
Reuben T. Harris, Chairman
Department of Systems Management

ABSTRACT

The purpose of this thesis is to propose and demonstrate a new negotiation and argumentation medium. This medium will take advantage of the latest in web technologies while conducting a detailed analysis and design of a prototype web based decision support system to support on-line argumentation, claims, and team decisions. The information obtained from the application will be stored in an ODBC database, to be used as part of the organizational memory. Organization memory will significantly enhance an organizations ability to utilize historical data in conjunction with current decision making requirements. The findings in this study strongly support the strengths of the action-resource based argumentation system (ARBAS) model and indicate that future research and application development would significantly advance the fields of web-based negotiation and argumentation. A web-centric prototype developed during this research can be viewed at [http://www.cimnet.nps.navy.mil/thesis].

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

A. AREA OF RESEARCH

The purpose of this thesis is to propose and demonstrate a new negotiation and argumentation medium. This medium will take advantage of the latest in web technologies while conducting a detailed analysis and design of a prototype web based decision support system to support on-line argumentation, claims, and team decisions. The information obtained from the application will be stored in an ODBC database, to be used as part of the organizational memory.

B. RESEARCH QUESTIONS

- What are issues related to supporting organizational memory?
- What are design considerations to support organizational memory?
- How can argumentation and claims be supported in a computerized environment?
- What technologies are available to support web based organizational repository?
- What is the relationship between technologies and how do they best support web based DSS applications?
- How does a hybrid information architecture better support DSS application design as it relates to RAD?
- What technologies are best suited for data warehousing, interface design, and reports as they are related to building an organizational memory support system.

C. BENEFITS OF STUDY

Early research in the fields of DSS to support organizational memory area has shown positive feedback. This benefit is somewhat limited due to the lack of integrated technologies to deliver the capability to the users. It is expected that the system will provide enhanced functionality that will ultimately increase the organization's ability to utilize historical data in an effective DSS for decision-makers today.

D. METHODOLOGY

To identify the required elements for the design and development of a web-centric negotiation and argumentation decision support system, we conducted a literature survey and analysis of existing web technologies. We explored various web-based technologies for rapid application development (RAD) capability. We preformed a comparison of the most advanced middleware products to determine which tool would best support the ARBAS-IOM (Action Resource Based Argumentation – Internet Organizational Memory) Web based prototype application.

To determine the feasibility of implementing an on-line Internet based negotiation application, we used the ARBAS model to develop a prototype. Allaire Cold Fusion was used as middleware software and Microsoft Access '97 was used as the relational database. ARBAS is a theoretical argumentation and negotiation language created by Dr. Tung X. Bui of the Naval Postgraduate School, Monterey, California.

In order to test the validity and utility of the prototype application, we conducted several iterations of tests and surveys with local students.

E. RESEARCH OUTPUT

A working prototype of the Internet based ARBAS-IOM application was created. The prototype is available for viewing at [http://www.cimnet.nps.navy.mil/Thesis].

F. ORGANIZATION OF THE STUDY

Chapter II discusses the history of computer supported group decision making, negotiation and argumentation. Chapter II explores the most current programs that are available for developing negotiations and decision making applications. This chapter will also discuss ARBAS, a language for supporting action-resource-based argumentation.

System analysis and design procedures for the development and implementation of the prototype application are discussed in Chapter III. The underlying framework for the application development was supported by the ARBAS model discussed in Chapter II.

Chapter IV discusses the operating procedures for the ARBAS-IOM application. This chapter includes a description of:

- Internet Organizational Memory (IOM) application.
- Procedures for accessing the IOM application.

Chapter V concludes the study by summarizing the research, providing recommendations, and discussing lessons learned.

II. LITERATURE REVIEW

A. INTRODUCTION

This purpose of this chapter is to describe the process of decision making, negotiation, and argumentation in the corporate environment. It also lays down the basis for generating corporate knowledge and organizational memory to aid in decision making. The Action-Resource Based Argumentation model and language is described in "ARBAS: A Formal Language to support Argumentation in Network-Based Organizations." (Bui, 1997) Feasibility is analyzed, as well as the costs associated with development and implementation of a decision making process using web based technologies.

B. COMPUTER SUPPORT FOR GROUP DECISION MAKING, NEGOTIATION, AND ARGUMENTATION

Decision making in the workforce can be a tedious process. Decisions are made with or without corporate knowledge. Corporate knowledge, otherwise known as organizational memory, is internally generated information (Ehrlich, 1994). Walsh and Ungson define organizational memory as " a construct that is composed of the structure of its information retention facility, the information contained in it, the process of information acquisition and retrieval, and its consequential effects." It is divided into two categories: "archival" which is information that will not change, such as manuals, product literature, financial reports, and presentations. The other information is "on-going", which describes things that change with time, such as business processes, budgets, discussion databases, and artifacts (Ehrlich, 1990). When corporate knowledge is unavailable, the decision-maker uses educational knowledge, experiential knowledge, or just plain gut instinct. Sometime that internal instinct or "SWAG" is not enough to make a well-informed decision. Most decision-makers have some corporate knowledge to make a lot of the decisions but not the depth needed for the process to be done correctly. Use of advanced information technologies leads to increased information accessibility, and this increase leads to improvements in decision making by creating more timely, comprehensive, and accurate organizational intelligence. Some decisions should be made on a group basis versus individual basis. When groups are involved varying degrees of corporate knowledge and expertise can be brought to the table. Group discussions can be limiting. Geography as well as personality can limit good decision making processes. This discussion will focus more on the geographical limitations than the personality limitations.

Geographical limitations revolve around the fact that the key players can not sit face-to-face around the same table and discuss the situation and come up with a solution to the problem. Other methods must be used since VTC's and conference calls may not be available. Web based applications lend themselves to distributed decision making. Use of advanced information technologies lead to increase in information accessibility, which leads to improvements in effectiveness of intelligence development and decision making. Whether it entails a company Intranet or the Internet groups of people can work together to solve a problem. Organizational memory accumulates over time and becomes valuable for informing new or distant employees and for enabling managers to make decisions based on past experience and knowledge (Ehrlich, 1994). The burden of the decision remains with one person but he can pool the knowledge and experience of others to make good decisions.

Corporate knowledge can contribute to the process in many other ways. When a group or individual makes decisions, those decisions and the processes are not very well documented. When a decision is reviewed at a later time, the process cannot be reviewed at the same time as the decision to help understand why a certain conclusion was drawn and a decision was made. Even if the decision-maker and or his staff are available, they may not remember all of the facts. Information technology has enabled businesses to generate and retain information (Ackerman, 1996). With this same Web Centric process corporate knowledge and process documentation can be generated from the beginning of the decision process all the way to completion. The goal-driven nature of organizations suggests than an organizational memory mechanism that is immediately tied to the on-going processes and considerations of the organization will be most important and useful to that organization (Ackerman, 1996). Web centric technologies allow users to conduct business and store that business in a database simultaneously. If an organization learns, then the results should be available for future use. The information stored in the database becomes organizational memory and is searchable. This data then can be used to understand past decisions as well as reference a past precedence to come up with an informed decision.

Gregory Kersten and David Cray at The Center for Computer assisted Management at Carelton University developed two negotiation applications called INSPIRE and INSS. INSPIRE is a web based negotiation support system. It contains a facility for specification of preferences and assessment of offers, an internal messaging system, graphical displays of the negotiation's progress, and other capabilities (Cray, 1994). INSS is very similar in its application. Both systems follow a sound approach to negotiation that has been proposed by experts and are the cornerstone of negotiation analysis. Cray and Kersten state that the three steps in negotiation are: preparation, the conduct of the negotiation, and post-agreement (decision).

Preparation for the negotiation is the time during which the user studies the situation, identifies the stakeholders, and develops a very clear understanding of the issues and interests involved. To help the user do this step, INSPIRE and INSS provide the user with a detailed description of the negotiation case and then guides the user through a sequence of pages on which he tells the system how important each issue and each alternative is. This step is also called preference elicitation. The information so obtained is used by both applications in the next step to give the user helpful feedback when constructing new offers or evaluating the counterpart's offers.

The actual conduct of the negotiation during which the user and his counterpart exchange a series of messages and offers, creating a suitable atmosphere for the negotiation, presenting the user's side of the case, and bargaining until the user reach an agreement. INSPIRE and INSS provide the user menus by which he can construct offers, and boxes for messages. Both applications further support the user by displaying a rating (score) beside each offer based on his preference information from the first step, and by plotting a graph of the history of both sides of the negotiation (entirely from the user's perspective).

The post-settlement period during which the user has the option of renegotiating an agreement that he has already reached. Based on the preference information provided by both the user and his counterpart, INSPIRE and INSS determine whether the agreement the user has reached is an "optimal" one in the sense that neither of the users can improve it without loss to the other side. (This is also known as a "nondominated" or "Pareto-optimal" agreement.) If INSPIRE and INSS can suggest better alternatives than the one the users have agreed upon, they are shown some of the possibilities and given the option of continuing the negotiation until they reach another agreement.

These applications and ARBAS-IOM are similar to decision support systems (DSS) except for the fact with ARBAS-IOM the assigned actors are making recommendations to solve a problem rather that a computer. After a problem is realized a member in the organization initiates a thread to begin solving the problem. He identifies the other personnel he wants to assist in the decision making process and they solve the issues at hand. ARBAS-IOM still allows for group discussion and individual opinions to assist the decision-maker decide on an appropriate solution. With a DSS the person trying to decide on an issue inputs information into an application and the computer lists alternatives to solve the problem. The person then chooses which alternative to use. The issue becomes how to stimulate objective discussion in problem solving and store the results for future use and analysis.

C. ARBAS: A FORMAL LANGUAGE TO SUPPORT ARGUMENTATION IN NETWORK-BASED ORGANIZATIONS

1. Introduction

Communication among team members has been widely recognized as a critical ingredient for effective organizational decision making and a way to alleviate coordination problems. In this sense, teamwork can be viewed as the conveyance of commitments between members to perform a task (Davis, 1983; Koo, 1988). Organizational members generate and exchange ideas, examine their validity and feasibility, assess their consequences, agree on a final action to be taken, and coordinate activities to implement the action. However, effective communication is difficult to achieve, while poor communication can obstruct team performance (e.g. Hackman, 1975). Another difficulty inherent to effective teamwork and communication is the lack of corporate knowledge that is crucial to support problem definition, problem analysis, argumentation and claims. In particular, when teamwork is driven by highly dynamic and often emotional negotiation, arguments are often convoluted, vague or incomplete, leading to less than optimal action outcomes.

This section proposes a formal language to support argumentation, decision-making and to preserve the argumentation process as organizational memory. First this thesis presents argumentation as a basis for organizational decision-making. Next, the rationale and principles of an action-resource based argumentation language are formally presented. Then the ARBAS formalism is described followed by an explanation of a computerized version in "A Computerized Version of ARBAS-IOM". Next, the use of the language is demonstrated with a case study based on the U.S.-Canada Softwood Lumber Dispute". Finally, the major contribution of this thesis work with respect to related research is presented.

2. Argumentation as a Basis for Organizational Decision Making

The literature on organizational decision making is unanimous in recognizing that a successful meeting is one that encourages divergence and radical thinking to ensure that all the possible alternatives are explored to enhance the chance of converging to the best solution. This search for a common solution is often the result of a continuous exchange of arguments and counter-arguments between participants (Axelrod, 1977; Hiltz, 1978, Sawyer, 1965). According to Flores et al. (for example, see discussion Chang, 1994; Lee, 1990), communication can be broken down into four primitive acts: *opening* (i.e., suggesting an action items), *negotiating*, *performing*, and *assessing*. This approach has been illustrated in a general business context by to supplement tradition workflow analysis with accountabilities of involved parties. To help promote the exploration and discussion of ideas, Toulmin et al. suggest the use of an argumentation language as a structured means to convey arguments, reasons, evidence, or assumptions. Perhaps the best known form of information exchange and reasoning is Hegel's dialectical language in which the *thesis* is challenged by the *antithesis* until a *synthesis* is found. This synthesis would capture and integrate all the critical actions that emanate during discussion. The term *argumentation* refers to the act of making claims, backing them up with supporting evidence or reasoning, criticizing or challenging those reasons (Ballmer, 1981). Ramesh and Whinston provide a comprehensive survey on work subsequent to Toulmin's.

For the purposes of this research, an *argument* is typically composed of assumptions supported by fact(s) or reasoning from which conclusions are derived. Assumptions and reasoning are backed by established experience or underlying knowledge. In a highly dynamic and often emotional decision making situation, arguments are often convoluted, vague or incomplete. When an argument or a decision is vague, significant costs could be incurred by the additional effort required to ascertain the true meanings of the language or to reduce the risks of misinterpreting that language (e.g., Haramundanis, 1992). Therefore, a concise and structured language might be desirable to promote reasoning, while controlling emotions (e.g., Ballmer, 1981; Bui, 1994; Thomas, 1992).

Furthermore, a structured language supporting argumentation is expected to force parties to explicitly reveal underlying differences in opinions. An example of a simple argument structure is proposed by Toulmin.   The argument structure consists of three elements: a *claim* that is used to express an opinion or assumption; *data* that support the claim; and a *warrant* that justifies the logical relation between the claim and its supporting data. Binbasioglu and Jarke use the construct *opinion*. According to the dictionary Webster, the word *opinion* connotes, however, a loosely reasoned, ill-structured conclusion thought out yet open to dispute. Chang and Woo propose another example of argument structure. In their protocol, a *sentence* is composed of two components: a *function* that is a category of speech-act; and a *content* that represents the domain knowledge used for the argument. According to Ramesh and Whinston, argumentation centers on positions, and a challenged argument requires appropriate defense and response from the individuals asserting the position.

Finally, the argumentation language should also be able to apprehend the evolution of problem representations over *time*, where the actions and the accompanying justifications of the stakeholders are to be explicitly represented; hidden agenda can interfere with reaching consensus.

3. An Operational View of Modeling Argumentation

a. An Action-Resource Approach to Problem Representation and Problem Solving

A decision problem can be seen as one that requires the decision maker to identify what *actions* to take and what *resources* are affected by these actions, given the restrictions (constraints) of the problem at hand. Actions necessary for solving a defined problem and the resources involved relate to each other as follows: Resources are generated or consumed as a result of actions taken, or conversely, actions require input resources and/or generate output resources.

We define the action and the resources which are *input* to the action and/or resources generated as *output* of the action as a unit and refer to that unit as an activity. In general, the activity is composed of an action which may take many resources as input and/or may generate many output resources. For representation purposes, the activity described by an *activity triplet* consists of:

input resources | action | output resources.

The activity triplet can be viewed as a "primitive" process component. An activity-triplet can represent a portfolio of more than one primitive action. It is important to relate resources to actions under argumentation, on which positions are taken. Although there are quite a variety of possible ways to link resources to actions, probing the impacts of a proposed action on resources increases decision makers' awareness of the relevancy of that action. More important, the irrelevancy of certain alternate actions, the idea of which appears appealing yet may turn out to be infeasible (Bui, 1995).

b. Decision Making Dynamics, Problem Evolution and Traceability

An activity can lead to another activity, thus supporting complex problem definitions and re-definitions (Bui, 1995). Borchardt suggests that new alternatives can be generated analyzing challenged propositions in terms of benefits and burdens, issues and interests: *Who* is to be benefited or burdened? *What* are to be the benefits or burdens? *When* are they to begin and terminate? *Where* are they to be in effect? How – organizationally and procedurally – are they to be effected? *Why* is enactment of the proposal in the organization's interest? Sebenius argues that evaluating current alternatives in the search for new ones helps set the limits of negotiation. Any new proposed action should not be convoluted, but rather should be convincing enough to gain acceptance from others. In this detailed search for alternate solutions with strong focus on evaluation, chances for concession/convergence are increased (Zartman, 1982).

The action-resource representation supports evolution of problem representation by *chaining* the activity triplets over time. The chained representation documents the dynamics of activities (or decision processes) during the course of negotiation. As the problem evolves, new (proposed) activities emerge to replace the old ones, and exchanged views on the activities give rise to new ones. In other words, the outcome of an activity can lead to another one, or a new activity can be introduced by the negotiators to reflect their (new) objectives and constraints.

Thus, the dynamics of decision making can be traced by navigating through the chain of activities that occurred during the project development process. Involved parties could navigate along the action chain and search for new solutions, until an action-resource triplet satisfies all parties.

c. *Toward a Formal and Operable Language for Dialectic and Evolutive Argumentation*

We use the word *argumentation* in a broad context as a means for communication, discussion, evaluation, proposition and decision. Keough and Lake note that arguments, as a discourse for negotiation, serve a distinctive instrumental function in that they are used for persuasion, information exchange, issue definition, and consensus seeking. The structuring of such a discourse can be represented as a "*view*" to support argumentation. We advocate the use of the word "v*iew*" as a general frame to include opinion, sentiment, belief, conviction and persuasion where individuals are able to substantiate their views by relating them to the relevant action. We adopt the theoretical framework proposed by Jackson and Jacobs. They posit the use of arguments as a conversational mode to support collaborative activities. An argument is not just a monologic process in which the "speaker" merely provides some reasons for his claims. Rather, it emerges when a proposed action is regarded as undesirable and further conversations and propositions are required.

Syntactically, a view can be in the form of opposition, agreement or no-opinion (or neutrality). Justification can be supportive or contradictory, and would relate the view to goals, constraints, facts or assumptions. Conversely, an assumption has the potential to be factual, but there is a certain amount of doubt (i.e., problematic fact). There is always a chance of switching back and forth between the class of assumptions and that of facts. When a challenged assumption cannot be defended, the problem components based on that assumption also lose support, and the problem structure is to be modified accordingly (McAllester, 1982)

4. ARBAS – A Language for Supporting Action-Resource Based Argumentation

We define a language to support argumentation with the following grammatical specifications: The *lexicon* seeks to provide a vocabulary for the argumentation discourse. Based on the lexicon specified, a *syntax* must be defined to derive a compound morphology of the argument. This refers to the grammar that deals with combinations of simple words. While the grammar looks at the correctness of the sentences, *semantic constraints* are used to impose a decision-making structure to the argumentation process. The following describes ARBAS formalism.

- **Lexicography:** An argument can be composed of a small lexicon uniquely composed of words that are necessary for an activity-resource-based discussion. We adopt a limited set of vocabulary to avoid convoluted arguments which seek to represent the negotiator's ultimate position.

  Given an action to be jointly accomplished (*Action_Name*) and resources to be used in accomplishing it (*Resource_Name*), each of the team members (*View_Owner*) expresses his position (*View_Position*) with personal inflection (*View_Intonation*). The position is justified by a discourse (*View_Justification*) with arguments on the possible impacts of the action being discussed on the resources (*Resource_Name; Resource_Type*). Discussion on the driven resources is implicitly related to the objectives of the task at hand, and explicitly on the amount of resources required for the tasks *(Resource_Quantity, Resource_Limit).*

- **Syntax:** An argument can be composed by following a simple set of rules:
  By combining words, a "view" can be derived.
  When "a decision maker expresses his/her view on the resource(s) related to the task at hand", a *"View_On_resource"* can be defined as follows:

  $$V_{\text{On-Resource}}(Z,RN,RT,RL)$$

  In the same manner, a *"View_On_Action"* can be composed to describe the situation in which "an owner, with his/her resource in mind, propose an action at time t":

  $$V_{\text{On-Action}}(V_{\text{On-Resource}},A,T)$$

  The position on the view and its *intonation* (i.e., inflectional morphology) can be interpreted using the words I and P: $V_{\text{Position}}(V_{\text{On-Action}},I,P)$
  A recommended action *("What_move")* can be specified using the constructs M and J which suggests an appropriate activity:

  $$V_{\text{What-move}}(V_{\text{Position}},M,J)$$

View_Owner:	Z = {z | z is a name of the team member}
Activity_Triplet:	A = {RN,RT | X | RN',RT'}
Action_Object:	O = {o | o is an object used to describe the action}
Action_Name:	X = {x(o) | x is a name of the action}
Resource_Name:	RN = { rn | rn is a resource name}
Resource_Type:	RT = {Input, Output}
Resource_Limit:	RL = { r,l | r ≥ 0; l is a resource limit {upper, lower}}
View_Intonation:	I = {Sentiment, Opinion, Belief, Conviction, Persuasion}
View_Position:	P = {Support, Oppose, Don't Care}
View_Justification:	J = { n | n is a free_format_argumentation based on goals, constraints, facts or assumptions}
Proposed_Move:	M = {Drop, Implement, Modify, Other}
Time_Stamp:	T = {t | t is a time stamp that covers all activities that occur between the start and the end of the process}

Figure 1. The Lexicon Notation

- **Semantic Constraints:** Constraints can be formulated to provide a minimum of semantic correctness to the argumentation language:
Argumentation Logic: This constraint implies that a position, P, should be followed by a logical move, M:

$$S = \{(p,m)|(\text{Support},\text{Implement}),(\text{Support},\text{Modify}),$$

$$(\text{Oppose},\text{Drop}),(\text{Oppose},\text{Modify}),(\text{Oppose},\text{Other}),$$

$$(\text{Don't Care},m^*)\}$$

where S is a constraint to validate the logic of argumentation, and $S \in (P \otimes M)$; m^* is a wild card. This logic also represents a transition for creating a new action when $m=\{\text{Modify, Other}\}$.

- **Operative Constraints:** A simple transition rule can be defined to assure argumentation continuity. A new action, x, at time $t+1$ will be proposed when, at time t, $s_t \in S'$:

$$\exists\, x_{t+1}, \in X,\ s_t \in S' = \{(p,m)|(\text{Support},\text{Modify}),$$
$$(\text{Oppose},\text{Modify}),(\text{Oppose},\text{Other})\}$$

Argumentation and generation of a new action x triggered by a move, m, could theoretically perpetuate with no convergence in sight. An operative constraint can be imposed by setting a *time limit* to terminate argumentation:

$$\exists\, V(t),\ t \le t'$$

where $t' \in T$, t' is a time threshold.

Another termination can be set based on the *maximum number of actions*:

$$\exists\ X,\ \delta(X) \leq c'$$

where δ is a cardinality function about a set, and c' is an action threshold expressed as a positive integer.

The third type of termination, and indeed, the most desirable operative constraint, is defined by a *consensus threshold*. The threshold, s', is a value with which members feel satisfied.

$$\exists\ x' \in X,\ V(z,x',p,i),$$

when $\Sigma\ (p,i)_Z \geq s'$ is met for $\forall\ z \in Z$.

The constraints described above are only illustrations of the use of the language for the organization to develop a context-sensitive argumentative reasoning and problem-solving formalism. New constraints can be formulated to satisfy the argumentation process and problem solving needs of a particular organizational situation.

5. A Computerized Version of ARBAS

a. ARBAS-IOM Systems Functionality

ARBAS-IOM provides a distributed platform that allows team workers to participate in a problem-solving setting without having to be physically present at a conference table. Temporal integration is assured with interactive queries searching for information about past negotiations. Involved members can navigate seamlessly through threaded discussions. As an organizational repository, it provides historical information regarding the decisions and discussions of the organization. The ability to search past history helps the negotiators gain a better understanding about the past decisions made in the organization. It also helps the decision-makers by providing new ideas and insights to make better decisions in the present based on information from the past

b. Implementation

A prototype of ARBAS-IOM has been implemented on a Microsoft Windows 95, Website Pro as the web server, Cold Fusion as the browser-database interface, and Microsoft Access 97. At any point in time, participants to an argumentation session can announce a new problem, generate propositions, specify their preferences, formulate arguments and interact with others. A process ends when one of the following conditions is met:

- Consensus is found.
- Agreement is reached by vote.
- Allowed time has elapsed.

6. An Example The U.S.-Canada Softwood Lumber Dispute

We use a well-documented case study described in Fang et al. and show how ARBAS-IOM could have been used to support and represent the dispute and its resolution. For the sake of clarity, the case study is briefly reproduced below.

The US-Canada dispute over the import of softwood to the United States started in May 1986 and ended December 30, 1986. In 1986, the U.S. lumber industry suffered a significant decline in the softwood lumber market. Meanwhile, Canadian wood cutters had been able to export US$2 billion/year of softwood lumber to the United States. The U.S. wood cutters argued that the reason that Canada enjoyed 30% of their market was due to the fact that the Canadian lumber industry benefited from low stumpage fees. In May 19, 1986, they formed the American Coalition for Fair Trade (CFT) and formally petitioned the U.S. Government (Department of Commerce) and the U.S. International Trade Commission (ITC) to rule on a charge of injury against alleged subsidized softwood lumber imports. CFT requested a duty of 27% on Canadian imports to offset the effect of alleged unfair trade.

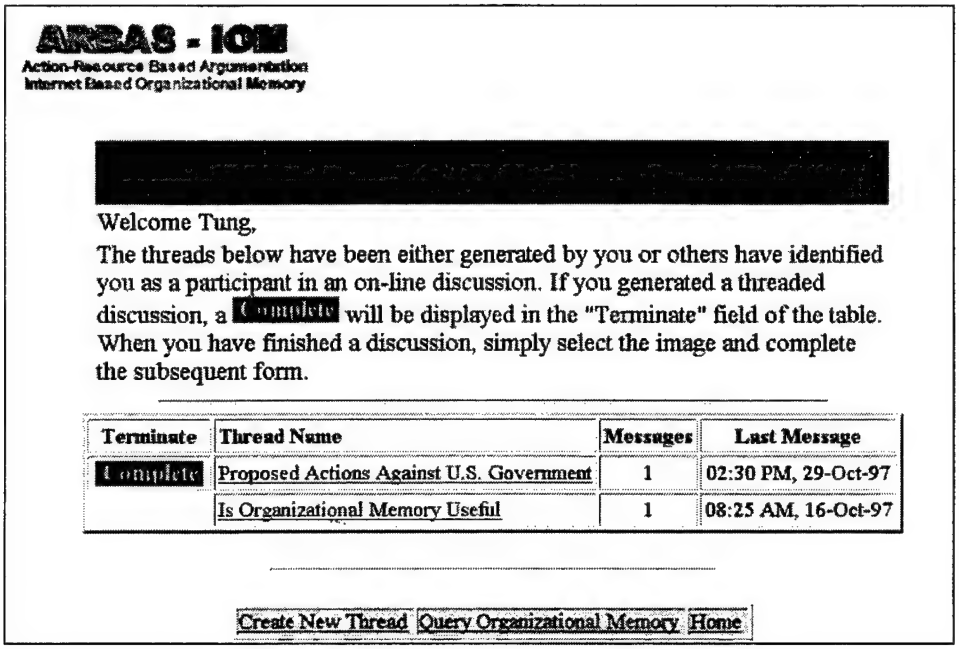
Based on its own investigation, ITC concluded on June 26, 1986 that softwood lumber imports from Canada were harming the American lumber industry. Subsequently, the U.S. Department of Commerce decided to impose a 15% import duty effective January 1, 1987. A few days after the U.S. import duty decision, the Canada coalition -- composed of various federal agencies -- provincial officials and lumber firms reunited to counter the U.S. decision. A number of actions were considered to include: campaign to protest against the U.S. ruling, voluntary restrictions on softwood export to the U.S. or increase stumpage fees to invalid CFT's petition. As a result of lengthy argumentation between various Canadian stakeholders, Canada decided on December 30, 1986 to raise the stumpage fee. The decision was justified by the fact that (i) there was no way to challenge the ITC decisions, and (ii) if the price of Canadian softwood lumber has to be raised by tax, it is the Canadian interest to raise its own tax. This Canadian decision, just a day before the U.S. tax would take effect, caused the CFT to withdraw its petition.

The problem is entered in ARBAS-IOM by an individual defining a new problem (which involves actors: notional Ministers of Finance, Commerce, and Foreign Affairs from Canada; triggering events; initial positions of the declaring party, and his propositions and arguments).

Figure 2 shows the user's screen after he has successfully logged into the ARBAS-IOM system. He sees any threads he has initiated as well as the number of threads that have been input for each problem. The user then chooses the thread he is interested in and responds to the thread based upon the problem, potential solutions and other actor's input. The next view will show the other actor's responses along with the initial thread.

Figure 3 shows a detailed view of the threads that have been added by all the actors. After reading the responses he is interested in the user can then add his own response to the problem. The user then chooses a position based on his views of the situation. He submits his solution along with what should be done, modify, drop, or leave unchanged.

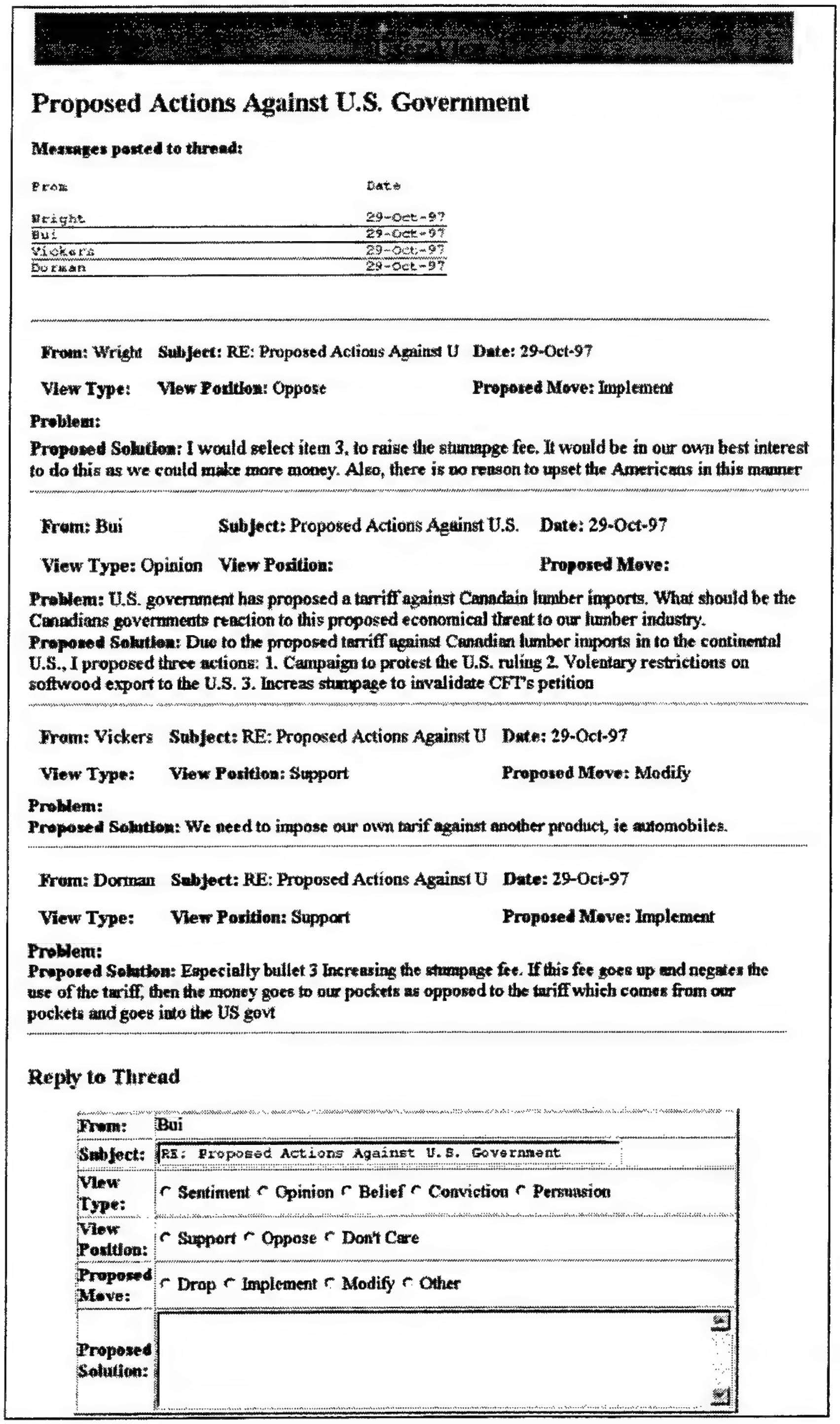
Figure 4 and Figure 5 illustrate how ARBAS-IOM could be used as a tool for organizational memory. The query searches for all the problems in which either Canada or CFT has been involved so far. As seen on the folders of the query window, search can be performed in using the following search keys: “keyword, subject or problem description, actors or involved parties.


ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory

Welcome Tung,
The threads below have been either generated by you or others have identified you as a participant in an on-line discussion. If you generated a threaded discussion, a Complete will be displayed in the "Terminate" field of the table. When you have finished a discussion, simply select the image and complete the subsequent form.

Terminate	Thread Name	Messages	Last Message
Complete	Proposed Actions Against U.S. Government	1	02:30 PM, 29-Oct-97
	Is Organizational Memory Useful	1	08:25 AM, 16-Oct-97

Figure 2. Main User Interface



Proposed Actions Against U.S. Government

Messages posted to thread:

From	Date
Wright	29-Oct-97
Bui	29-Oct-97
Vickers	29-Oct-97
Dorman	29-Oct-97

From: Wright **Subject:** RE: Proposed Actions Against U **Date:** 29-Oct-97

View Type: **View Position:** Oppose **Proposed Move:** Implement

Problem:

Proposed Solution: I would select item 3, to raise the stumapge fee. It would be in our own best interest to do this as we could make more money. Also, there is no reason to upset the Americans in this manner

From: Bui **Subject:** Proposed Actions Against U.S. **Date:** 29-Oct-97

View Type: Opinion **View Position:** **Proposed Move:**

Problem: U.S. government has proposed a tarriff against Canadain lumber imports. What should be the Canadians governments reaction to this proposed economical threat to our lumber industry.

Proposed Solution: Due to the proposed tarriff against Canadian lumber imports in to the continental U.S., I proposed three actions: 1. Campaign to protest the U.S. ruling 2. Volentary restrictions on softwood export to the U.S. 3. Increas stumpage to invalidate CFT's petition

From: Vickers **Subject:** RE: Proposed Actions Against U **Date:** 29-Oct-97

View Type: **View Position:** Support **Proposed Move:** Modify

Problem:

Proposed Solution: We need to impose our own tarif against another product, ie automobiles.

From: Dorman **Subject:** RE: Proposed Actions Against U **Date:** 29-Oct-97

View Type: **View Position:** Support **Proposed Move:** Implement

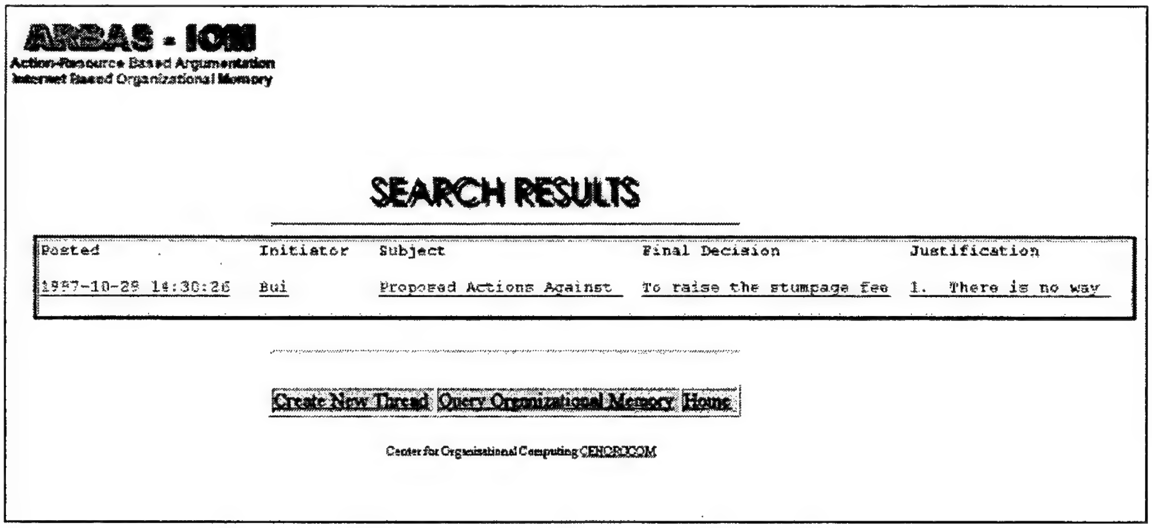
Problem:

Proposed Solution: Especially bullet 3 Increasing the stumpage fee. If this fee goes up and negates the use of the tariff, then the money goes to our pockets as opposed to the tariff which comes from our pockets and goes into the US govt

Reply to Thread

From:	Bui
Subject:	RE: Proposed Actions Against U.S. Government
View Type:	○ Sentiment ○ Opinion ○ Belief ○ Conviction ○ Persuasion
View Position:	○ Support ○ Oppose ○ Don't Care
Proposed Move:	○ Drop ○ Implement ○ Modify ○ Other
Proposed Solution:	

Figure 3. Thread Discussion and Response



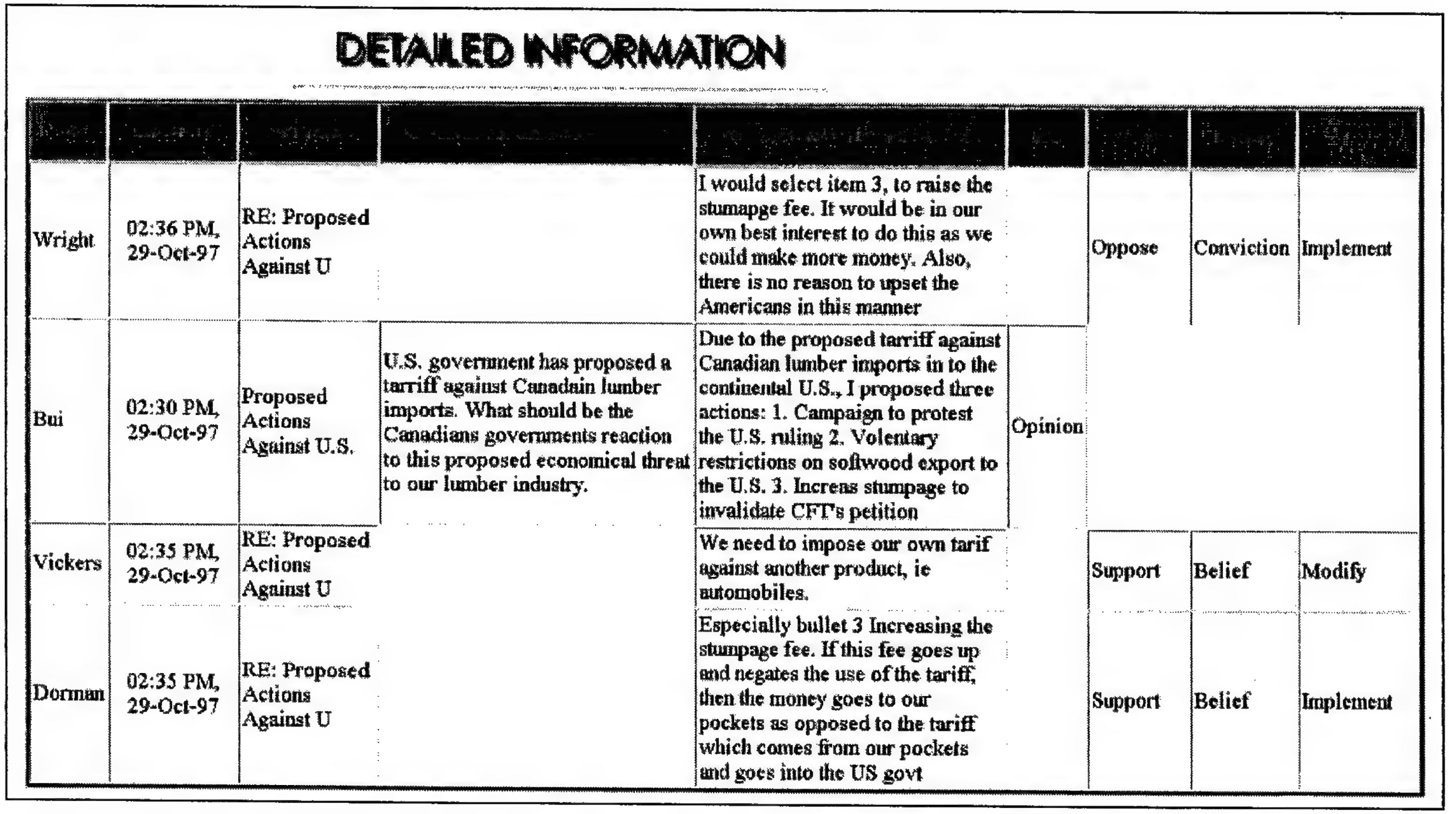
ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory

SEARCH RESULTS

Posted	Initiator	Subject	Final Decision	Justification
1997-10-29 14:30:26	Bui	Proposed Actions Against	To raise the stumpage fee	1. There is no way

Create New Thread
Query Organizational Memory
Home
Center for Organizational Computing CENORCOM

Figure 4. User Query



DETAILED INFORMATION

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Wright	02:36 PM, 29-Oct-97	RE: Proposed Actions Against U		I would select item 3, to raise the stumapge fee. It would be in our own best interest to do this as we could make more money. Also, there is no reason to upset the Americans in this manner		Oppose	Conviction	Implement
Bui	02:30 PM, 29-Oct-97	Proposed Actions Against U.S.	U.S. government has proposed a tarriff against Canadain lumber imports. What should be the Canadians governments reaction to this proposed economical threat to our lumber industry.	Due to the proposed tarriff against Canadian lumber imports in to the continental U.S., I proposed three actions: 1. Campaign to protest the U.S. ruling 2. Volentary restrictions on softwood export to the U.S. 3. Increas stumpage to invalidate CFT's petition	Opinion			
Vickers	02:35 PM, 29-Oct-97	RE: Proposed Actions Against U		We need to impose our own tarif against another product, ie automobiles.		Support	Belief	Modify
Dorman	02:35 PM, 29-Oct-97	RE: Proposed Actions Against U		Especially bullet 3 Increasing the stumpage fee. If this fee goes up and negates the use of the tariff, then the money goes to our pockets as opposed to the tariff which comes from our pockets and goes into the US govt		Support	Belief	Implement

Figure 5. User Detailed Query Results

7. ARBAS: Related Work and Contributions

Toulmin's argumentation structuring approach has been the inspiration for many recent formal systems which attempt to promote opinion exchange and joint decision making. Similar to our design objective, the systems cited above assume that agents have individual and common goals. Decision-makers are encouraged to interact according to well-defined communication protocols. Opinions and ideas are solicited, consolidated, and supported by facts and arguments, so that counter-productive, rhetorical or circular arguments are progressively eliminated, and a central and collectively agreeable issue eventually emerges. Conversely, ARBAS as a formal language is context-independent. Also, the representation languages of the cited systems are not powerful and flexible enough to capture the dynamics of negotiation. The dialogue management is not sufficient to guide participants to take actions (either pro-actively or reactively) and does not provide a framework for stakeholders to quickly estimate the impacts of proposed activities.

Most existing systems act as a representation or snapshot of the situation, thus lacking the ability to *time* the sequences of activities. Our representation language allows involved parties to trace back and forth various routes, thus allowing simulation, search for new alternatives, and backtracking.

From an operational viewpoint, ARBAS provides a number of unique contributions:

- **Negotiation support:** Negotiation can be costly and counterproductive if poorly handled. Organization effectiveness can be seriously compromised if teamworkers hide their disagreement and avoid confrontation. Supporting explicit negotiation task with a very simple yet powerful language to express user opinion effectively addresses this issue. ARBAS provides an innovative way of negotiating. Parties negotiate by taking into account the most essential viewpoint: the input and output resources that are linked to the proposed action.

- **Coordination:** As ARBAS is task-oriented, it facilitates coordination among co-workers. An opinion expressed by a member on a given action is a commitment, as the member is fully aware of the implications of his commitment for his resources.

- **Support for asynchronous working mode:** Problems related to the availability of people for face-to-face meetings are well-documented. ARBAS lets people organize their professional interaction at their own convenience. Operative constraints supply a means for imposing possible delays. For example, a member fails to return his input by the deadline at the end of the delay, other participants could assume that he/she does not care. ARBAS works as a way to synchronize the entire project without increasing project management workload.

- **Documentation as a means to preserve corporate memory:** Documenting all activities of an organization during its lifetime is indeed a costly and tedious task. To allow decision-maker to focus on solving a task at hand, a secretary is usually assigned to take notes manually and to produce minutes of the meetings. The quality of the documentation (e.g. minutes of meetings) depends on the skills of the note taker. ARBAS provides an automated and structured means to record the discussion. The information can be retrieved by using *Action* to trace the past decisions and their rationale.

8. Conclusion

In this paper, we propose an argumentation language as a conversational medium for members of a networked organization. The language is designed as a formal means that can be used to promote communication, organizational decision making and be used as a computerized organizational repository. The decision making process can be viewed as the identification of appropriate actions and resources, deliberation of organizational goals, and acknowledgment of constraints which might impede the consideration of certain actions.

A decision problem can be seen as one which requires the team members to decide what action to take and to identify what resources affected by this action. From a communication angle, we view *decision makers' interaction* as a series of propositions and counter-propositions of different tasks related to the software development project. Proposals and counter-proposals represent binding actions in which members are aware of the consequences of the proposed actions in terms of effort required to achieve the tasks and the benefits. We further regard team interaction as an iterative problem-solving process in which the task at hand is continuously defined and *re*-defined until a satisfactory solution is found. When an activity is planned, agents can question/oppose its existence, support its implementation, request modification, or suggest alternative activities. Satisfaction among members implies that expected results (goals) as well as resources required to achieve the task are agreeable to all. In this context, we use the word *argumentation* to depict communication, discussion, evaluation, proposition and decision.

We propose ARBAS as a structuring language that incorporates an argumentation scheme to present ideas that would likely stimulate focused discussions among the decision-makers. We argue that using an action-resource approach, workers are forced to defend their interests and take actions, by making clear their views regarding a (binding) action or move. The explicit representation of the relationships among the problem components also makes it possible to identify situations of likely conflicts when more than one party wants to share the same resources or wants to set their desired levels. As such, the proposed approach could be used as an effective means for promoting collaborative work. Since the proposed approach imposes structure on the discourse of views, it can also be employed when consolidating individual views into a group's view (using a brainstorming technique for example).

C. CURRENT TRENDS IN DECISION MAKING AND NEGOTIATION

1. Feasibility

Part of the development of an application is the determining the feasibility of such an application. The purpose of a feasibility study is to investigate and assess the degree of risk associated with developing and using a negotiation system. Three types of feasibility are: Technical Feasibility – can the proposed system be implemented with the currently available hardware and software; Economic Feasibility – whether the proposed benefits of the solution outweigh the costs of the application; and Operational Feasibility – is the proposed solution desirable within the current managerial framework of the organization.

Technically, a web centric decision making/discussion application that can store this corporate knowledge is very feasible. Information technology has enabled organizations to generate and store enormous amounts of organizational information (Ackerman, 1996). With the emergence of the World Wide Web and its off shoot technologies, distributed decision making is quite possible. There is no need for huge main frames with large databases since the personal computer has become so powerful. Network technologies have emerged as the forefront of distributed processing applications allowing users to communicate via the Internet, an Intranet, and/or an Extranet. The only technical challenge to such applications is the capturing of non-written discussion and information. Informal negotiations at times are done in a non-business atmosphere such as the golf course, luncheons, and telephone conversations, as well as, formally via VTC's, conferences, and meetings. How do organizations capture and store such information for future reuse.

Economically an application such as ARBAS-IOM and INSPIRE are quite feasible. The cost to implement such technologies has dropped greatly over the recent years allowing more people to use distributed applications. The decrease in computer costs, network hardware costs, and software costs is the reason for this availability of web centric technologies to more users.

Organizations have always had a problem with keeping track of information flow and decision making. The company reviews decisions and policies and cannot always determine who made the decision, why the decision was made, or how the decisions maker came about making the decision. From an operational standpoint a mechanism to analyze the decision making process would be beneficial. The only set back is, as designed, ARBAS-IOM does not allow for anonymity in the discussion as a group decision support system would. Anonymous discussions would allow for more honest input from the users. Since this is not designed to be a GDSS it is not a major issue. Changes in the design could be made if there was such a requirement, but that would add to the complexity of the application.

A feasibility study should be conducted in each organization prior to implementation of any application. This will aid the organization in defining the costs for implementing the application as well as determining the requirements needed to implement such an application.

2. Cost Issues

> A decision making process, like any other activity, has its own attributes, including effort, time, commitment and cost. A rational decision-maker will not embark on a process without the expectation that the results will outweigh expenditures. This is a key issue and all too often ignored in decision analysis although postulated a long time ago by Simon (1960) and others. (Kersten, 1996)

Negotiation Support Systems (NSS) like ARBAS-IOM generally are not found as off the shelf systems since this is a relatively a new application. These systems come in various styles. INSPIRE, INSS, and Answer Garden are examples of systems that are being developed for negotiations. Each system has different uses, with the common thread of negotiation, decision making, and storing the information gained through the decision making process. No specific cost for the INSPIRE or INSS applications could be found from Internet. Depending on complexity the cost of developing a system in house could be cheaper than an off the shelf application. ARBAS-IOM was a relatively inexpensive system to develop. Discussion of the actual application will follow Chapter IV.

The application was developed to use web technologies to conduct decision making on an Intel based machine. The costs for development can be broken down into fixed and variable costs. Some of the fixed costs are the hardware and the software with which the application was developed. The development of the application was conducted on an Intel Pentium based machine that cost approximately $2500.00. The software used in the development of ARBAS-IOM was Allaire's Cold Fusion 3.0 a CGI middleware replacement for database/browser interaction. Netscape Communicator 4.03 was used for testing and viewing the application. Microsoft Access97 was the ODBC compliant database used for storing the organizational memory. Hot Dog a web page development tool was used to develop the actual web site. A web server was also needed to stage the web site. O'Reilly's Website Pro 1.0 was chosen as the web server. The cost for the software is approximately $1500.00. Labor, a variable cost, was the most expensive portion of the project. The application prototype was developed in approximately 200 person hours. This figure covers all of the life cycle up to testing, implementation, and maintenance. The development labor cost would be approximately $12,000.00. This cost would be significantly higher if the project was taken to completion, beyond just a prototype. Each organization that wanted the application would have different requirements and design specs above the basic core application. These requirements and specifications would change the variable costs of the application. Due to the uncertainty of the changes it would be difficult to estimate the total labor costs. The approximate cost for in house development is $16,000.00. This cost only provides a useable prototype. In other words, an organization could implement ARBAS-IOM without any modifications in its current design. Designing a fully customized application, though expensive, is the recommended solution when the decision making process is relatively unique and there is no system available that lends to open discussion, decision making, and storage of the process.

To implement the application as is, as company would need to have a network with a web server, a web browser, and Cold Fusion. Costs for this could range from a few hundred dollars for the software to many thousands of dollars for hardware and software. The latter would depend on the size and existence of the network. Training the users would need to be included in the cost of the system that is implemented. The cost of the train would vary with the complexity of the application.

3. Web Based Decision Making

The Internet or web based technologies bring a great technology to decision making. It allows users in an organization who are not geographically close together to discuss departmental as well as company issues. These discussions turn into a wealth of knowledge for the decision-maker to make a good decision. Outside reference material is also accessible if the browser is used on a computer that has access to the Internet. During the process, any user can query the database to see if this issue has been discussed or a decision has been made on similar issues. This will aid the user's ability to provide an informed solution to the decision-maker so he can strong decision.

The browser also provides a great graphical user interface. The user is guided through the process with each page. As he answers questions and submits the results the browser sends him to the appropriate page to continue the process. Browsers are also multi-platformed. A UNIX machine views HTML with a browser the same way as an Apple machine does. This allows for greater interoperability between departments and organizations. The application can be accessed from anywhere the user is as long as he can access the web server.

Time is also better utilized using web based technologies. Discussions can be held in near real time. There is no need to send faxes, memos, letters, etc.. Because the input from other users is so fast, a decision can be made quicker. Since the application can be accessed from any location, input from users on travel or on vacation can be obtained. The decision maker does not have to wait for another player to return to complete the process. The input to the database occurs automatically. Every time a thread and a response are submitted they are stored in the central repository for future access and review. This save time also because another person does not have to manually input the data.

There are a few disadvantages to the system. Since the application requires a user to log-in some one must provide server access. This is a fairly easy problem to overcome since most networks employ some type of security which allows its users to log-in from remote locations. For the process to be timely the user must log-in on a regular basis. Like checking e-mail, the user must log-in to the server to keep abreast of the issues he is working on. An organizational policy could be established to help this problem. Lastly, with any busy network, bandwidth and reliability could be an issue. If the network is down users cannot keep up with their input. This application relies heavily on a dependable network for its success.

III. METHODOLOGY

The ARBAS model, introduced in Chapter II, was used as the framework for the design and development of the Internet Organizational Memory (IOM) application. ARBAS is an iterative process in that each actor/participant provides feedback to one or more previous actors/participants for the purpose of negotiation and consensus building discussion. The following system development life cycle (SDLC) was used to develop IOM:

- Analysis
- Design
- Development
- Implementation
- Evaluation

The purpose of this chapter is to describe the process involved in applying the ARBAS model to the ARBAS-IOM prototype application.

A. ANALYSIS

1. Industry Trends

For the Internet to emerge as a serious application environment, organizations will need to expand their development toolkits. In many reviews for IS and application development managers, the central topic of discussion today is middleware, database development tools, APIs, and deployment tools. There is a strong feeling in the market place that the traditional legacy enterprise solution providers will be overpowered by the combination of 4th generation languages, such as VB 5.0 and Java, and middleware applications. Particularly since many of the smaller Web technology companies are developing software significantly faster than mainstay solution providers. This is also supported by the fact that many of these smaller companies are merging to provide extended value added services. The real question becomes the nature of data storage, being the software platform it resides in, and picking the proper format to deliver the information in a timely and integrated fashion. A large number of organizations are using middleware to extend their organizational information infrastructure to exterior and interior agencies. It would seem that middleware technologies provide a robust and economical vehicle to accomplish this.

2. Determine the Rapid Application Development (RAD) Tools

The concept for the IOM application was unique in the sense that it is able to integrate the functionality and performance of a client/server based application without the traditional limiting geographic boundaries. For this reason, web-centric RAD tools were evaluated based on: ease of use, cost, capability, and the portability of the final product.

3. Determine Application Objectives

The application was designed to support iterative, on-line negotiation and argumentation. The desired outcomes were to:

- Introduce the actor/participant to the ARBAS-IOM application via hypertext documents.
- Provide the actor/participant with an overview of the capabilities and functionality of the ARBAS-IOM application.
- Enhance the actor/participants ability to make timely and informed decisions in a web-centric forum based environment.
- Provide the capability to track and ascertain the perspectives behind specific historic organizational decisions.

B. DESIGN

1. Application Design

The basic foundation of the IOM application was interpolated from the ARBAS model defined in Chapter II. The goal was to assimilate as much of the model as possible into a web-centric application. Careful consideration was given to designing interactions that would not confuse, frustrate or insult the user. Significant considerations were also given to the interface design of the application, maximizing screen functionality while minimizing congestion.

2. System Considerations

The following system considerations pertain to the server-side workstation and application development platforms. The server-side system configuration was difficult to ascertain as the web-server could provide additional services for much more than the ARBAS-IOM application. In this case, the web-server doubled as the application development workstation. To this end the following is considered to be the minimum configuration:

- Pentium 200 CPU
- Minimum of 900 Mb of free hard disk space
- 64 Mb RAM
- Super VGA video card with monitor capable of displaying 800x600 resolution
- Mouse

In order for an end-user or actor/participant to use the application, the following minimum system configuration would be needed:

- 486 compatible CPU
- Minimum of 30 Mb free hard disk space

- 16 Mb RAM
- Super VGA video card with a monitor capable of displaying 640x480 resolution
- Netscape or Internet Explorer Web Browser software
- Access to the Internet

3. Software Considerations

Several development tools for creating Internet based applications were considered. An authoring tool, web server software, graphics program, database software and middleware software.

a. Authoring Tool

The authoring tool was used to create the web-centric application templates as well as the hypertext mark-up documents. The selection of the tool was based on the following:

- Ease of use
- Whether it provided the required functionality
- Cost

There are so many tools available that it becomes increasingly difficult to test and evaluate all. The web-centric application templates are text orientated in nature and do not require substantial WYSIWYG (What You See Is What You Get) functionality. Developing these templates is akin to writing a program in Perl. A common word processor program would provide the desired results.

The HTML documents have the advantage of advanced layout and design capabilities. Significant consideration is given to cost and performance. WYSIWYG becomes increasingly important the more advanced the application structure becomes. Many tools provide advanced functionality that can significantly enhance the end-users satisfaction with the application interaction.

b. Web Server Software

The web server software selected should provide the basic functionality required by the application. There is a direct correlation between cost and performance of the web server software. Traffic analysis and forecasting will ultimately determine how robust a server is required.

c. Graphics Program

The graphics program should allow for the manipulation of all known graphic formats. Advanced functionality such as image mapping, transparency, and image conversion should be important considerations in final product selection. Many of the most useful programs are available as shareware on the Internet. While these do not always offer an enterprise solution for all of the graphic requirements, many can be met for little or no cost. Finely, the program should allow the user to save the end product to any of the popular file formats.

d. Database Software

The database must be ODBC compliant. This will insure that the middleware application software can interact with it. In many instances, a relational database can significantly enhance the performance of the application. Consideration should be made regarding the following:

- Cost
- Ease of use
- Performance
- The databases ability to interact with other non-proprietary software applications
- The number of simultaneous sessions the database can support at one time
- Security

e. Middleware Software

Most organizations use a database in one form or the other. The primary purpose of a database is the storage of valuable corporate information. Each department within the organization may posses an autonomous data base which stores information that is pertinent to the mission that it pursues. For instance, the sales department may store information regarding regional sales figures or product specific information. The human resource department may store information pertaining to employee benefits or performance. The purpose of Intranet based technologies is to provide a means to view, manipulate and use that information in a manner that is value added to the organization. Value added benefits can come in many forms.

- Providing sales representatives valuable information when they are away from the office.
- Enabling employees to view and change their benefit information on-line.
- Allowing managers to create on-the-fly reports.
- Proving sales reps with up to date product information and real time inventory levels.
- Provide the capability to better manage large document based information repositories.
- To unlock the potential of unused information held within organizational databases in support of identified mission objectives.

There are several ways to query and input data into a database from a web page and browser. CGI, or Common Gateway Interface, is the most widely utilized mechanism to support these processes. However, CGI requires the user to be an adept programmer. There are several companies that have realized the need for products that provide the functionality of CGI scripting without having to do it. This kind of software is commonly referred to as middleware.

This section provides a basic introduction to middleware software packages. It describes middleware capabilities and features. Web-centric middleware software can significantly enhance Web application development, allowing for the creation of dynamic-page applications and interactive Web sites. Middleware software provides developers a way to quickly build powerful Web applications that integrate with key server technologies such as relational databases and SMTP e-mail.

Because many of the middleware software packages are RAD, developers are not required to learn new programming languages or required to install additional client-side or browser components. Instead, developers create applications by combining standard HTML files with CFML (Cold Fusion Markup Language) tags that are easy to understand and use.

Web developers use CFML to build a wide variety of Intranet and Internet applications including:

- Customer Feedback
- Personnel Management
- Online order entry
- Event registration
- Online catalogs, directories, and calendars
- Bulletin board-style conferencing
- On-line technical support
- Internet commerce solutions

In a normal Web site, pages are simple text documents marked with HTML. The Web server sends out these pages to the user's browser as they are requested. A middleware Web application begins with the collection of dynamic-page templates instead of static HTML documents. A template is a simple text file that contains both HTML and the Cold Fusion Markup Language (CFML). Instead of being sent directly to the user's browser, templates are pre-processed by the Cold Fusion Application Middleware Server which generates an HTML page that is then sent to the user's browser. Figure 6 shows what happens when a Web browser requests a Cold Fusion Page.

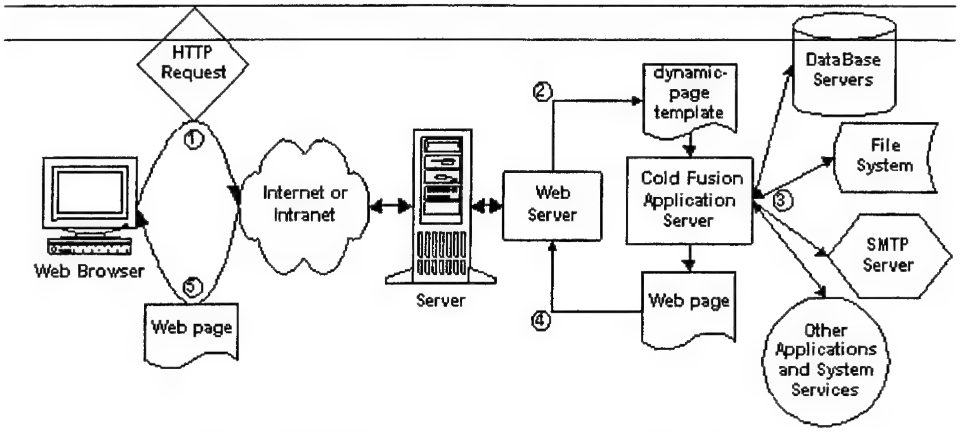


Figure 6. Cold Fusion Application Server

C. DEVELOPMENT

The middleware software is the heart of the IOM application. Allaire's Cold Fusion 3.0 was selected to provide the interaction between the Web pages and the Microsoft Access '97 database. The Cold Fusion engine allowed the IOM application to insert, update, delete, and perform advanced queries on the database. Access '97 was selected because of its ease of use, cost, and ODBC compliance.

1. Decision Support System Application Development

As discussed previously, the IOM application was developed utilizing the ARBAS model discussed in Chapter II. In order to capture the functionality of the model, significant consideration was given to the information architecture of the IOM application, as well as, the personal interaction of the user during each on-line session. Figure 7 illustrates the conceptual story-board and data flow of the IOM application.

Each of the boxes in Figure 3.2 represents a coded module in the application. Table 1 describes the basic functionality of each of the ARBAS-IOM code modules depicted in Figure 7.

The code modules for the IOM application were developed utilizing the Microsoft Word text editor. In order to provide the web based functionality, both HTML and CFML programming languages were used. HTML provided the basic web-centric document structure. CFML provided the embedded SQL to interact with the Access '97 database.

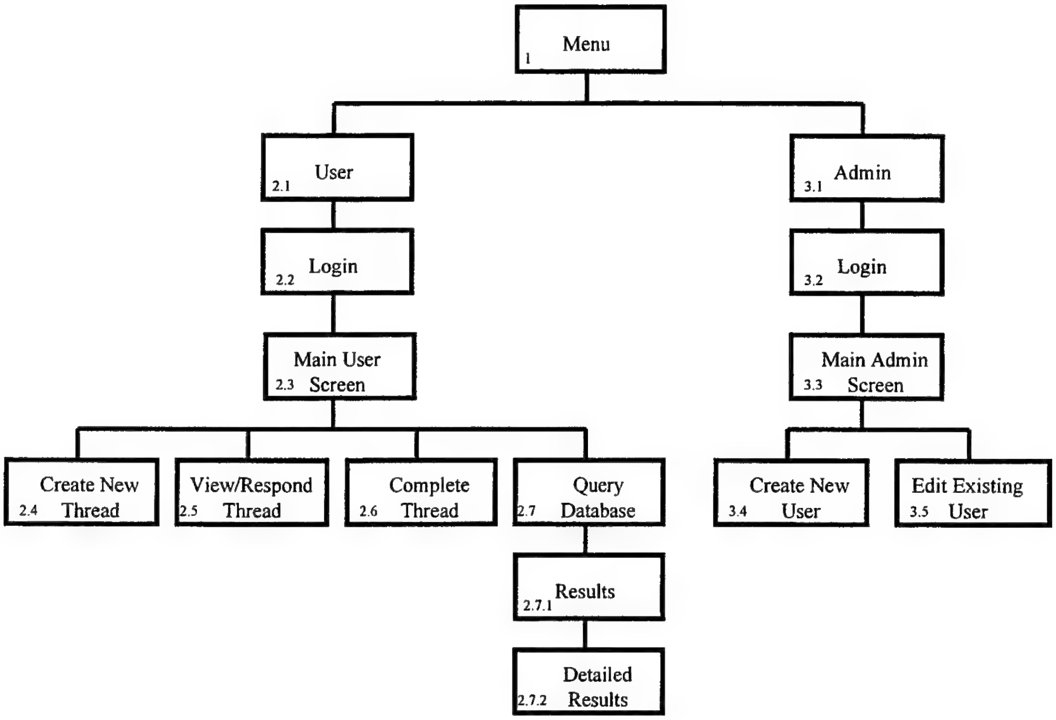


Figure 7. ARBAS-IOM Story-Board Application

Cold Fusion 3.0 provided a graphic user interface (GUI) wizard to assist in generating CFML code. This would constitute the RAD portion of the application development tool. The wizard provided the ability to query and insert data into an ODBC database. However, the RAD functionality ended with those basic operations. All additional advanced functionality required detailed programming in a text editor.

Table 1. ARBAS-IOM Coded Module Descriptions

Figure	Name	Function
1	Menu	Allows a user to select either User or Admin application functionality's
2.1	User	Provides the form for the user to specify last name and password
2.2	Login	A script file that validates the users or denies access if incorrect login attempt
2.3	Main User Screen	Displays all current on-line discussions, provides menu for 2.4 – 2.7 functions
2.4	Create New Thread	Allows the user to initiate a new threads negotiation or discussion
2.5	View/Respond Thread	Displays all of the messages that other users have sent, allows user to respond
2.6	Complete Thread	Allows user to complete threaded discussion and document final decision
2.7	Query Database	Allows user to query database based on keywords or user last name
2.7.1	Results	Displays a concise list of results
2.7.2	Detailed Results	Displays a detailed list of results
3.1	Admin	Provides the form for the Administrator to specify last name and password
3.2	Login	A script file that validates the Administrator or denies access if incorrect login
3.3	Main Admin Screen	Displays all current users, provides menu for 3.4 – 3.5 functions
3.4	Create New User	Allows the Administrator to create a new user in the IOM application
3.5	Edit Existing User	Allows the Administrator to modify the users database information

The design of the IOM application was directly influenced by the nuances associated with web application development. The IOM application is required to function in a static environment, creating many conditional-coding requirements. Each transaction by the user calls specific modules of code, stored on the web server, to provide the desired result. The application is required to anticipate what the user will want to do next and provide that functionality.

2. Database Development

Database design was directly influenced by the ARBAS model in Chapter II and the web-centric nuances associated with the IOM application. Tables were created to support the identification and tracking of negotiations/discussions. For the purpose of this application, each new negotiation/discussion was labeled as a Thread and stored in the Threads database table. Each new response, from an actor/participant, is labeled as a Message and stored in the Message database table. Additional fields were created to support Internet application execution. Examples of these fields would be for Cookies and time/date stamping.

The database is not entirely relational, only the Threads and Message tables. Because of the modified SQL code involved in the CFML programming language, it was extremely difficult to insure that database integrity was maintained. In order to insure integrity, table relationships were kept to a minimum.

D. IMPLEMENTATION

The next step was the development of the actual IOM application. The basic goal was to provide users with a Web based interface that would enhance the negotiation/discussion process and capture all relevant information for future use. The design scheme provided the basis for IOM application development in the web-centric environment. All transactions are completed utilizing Internet based technologies. The primary advantage of this is the portability of the interface applications. Because they (documents, applications and database) reside on the web server, a client can be of any operating system and still interface with the database via TCP/IP and HTTP protocols.

1. Application Implementation

As alluded too earlier, the IOM application is comprised of web pages. These web pages consist of various different programming languages. HyperText Markup Language (HTML) was used to present many different types of media to include text and graphics.

NetObjects Fusion was used to build the applications static web pages. This program provides the capability to design and publish entire web sites vice creating one page at a time. NetObjects Fusion is a total WYSIWYG development environment. As text, graphics, or multimedia objects are placed on the screen, they appear in the browser.

Cold Fusion was used to handle the middleware requirement of the application. Cold Fusion Markup Language (CFML) was used to imbed SQL into the HTML web pages. The web server processes the CFML as HTML, passing the code to the Cold Fusion Engine for the interaction with the database. The Cold Fusion Engine also provides dynamically generated web pages as a result of a user query to the database. CFML module code was generated in a Microsoft Word text editor.

E. EVALUATION

The final phase of the SDLC involved the use of a survey instrument to evaluate the IOM application. The survey was administered via a web page and the data was stored in an Access '97 database. The purpose of the survey was to measure the user's perception of the utility of the application and to identify any possible application shortfalls.

IV. OPERATION OF THE IOM DECISION SUPPORT SYSTEM APPLICATION

A. INTRODUCTION

The primary purpose of this thesis was to create a web-centric negotiation and argumentation decision support application. The secondary purpose was to identify and measure the effectiveness of Internet technologies in support of on-line negotiation/argumentation. The ARBAS model was chosen to provide the foundation for the business processes of the IOM application.

This chapter will discuss the IOM application, procedures for accessing and running the IOM DSS application, and system requirements.

B. INTERNET ORGANIZATIONAL MEMORY APPLICATION

The IOM application consists of twenty-two Cold Fusion Metafiles (.cfm) and an Access '97 database file (.mdb). Each of the Cold Fusion Metafiles represents a module, which provide the various functionality's to operate the IOM application. The modules have the capability of individually interacting with the database file, depending on the functionality the user is trying to exploit. The modules are maintained in the root directory of the web application server. From this location, any user can access the IOM application from a workstation with TCP/IP connectivity and a Web browser.

Figure 8 depicts the modules architecture for the IOM application. Certain module operations will return the user to the main user screen. The dotted line and arrow illustrate this. This line is also meant to indicate that the ARBAS-IOM application updates during this evolution, displaying the most current information to the user.

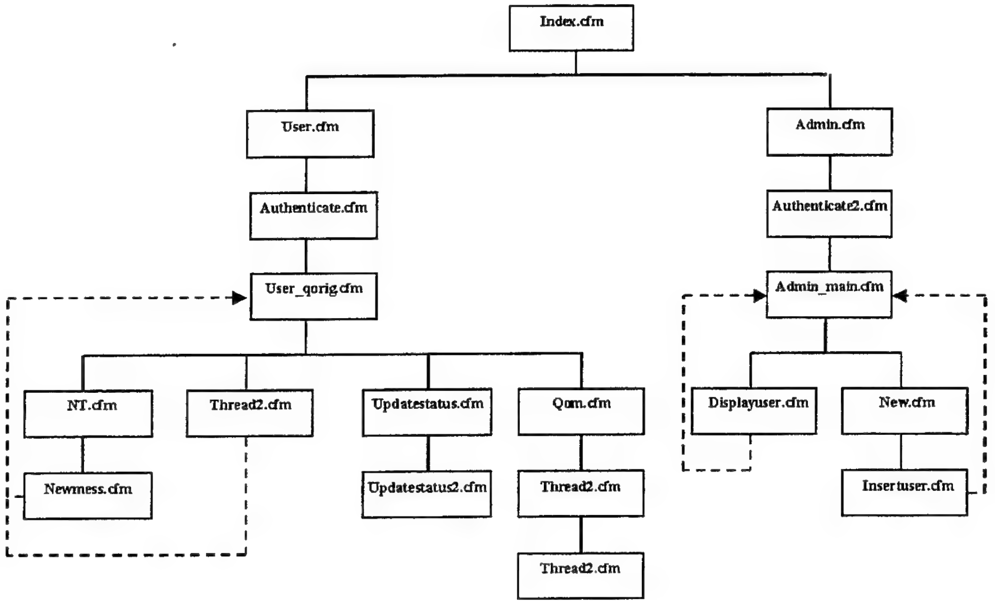


Figure 8. ARBAS-IOM Module Architecture

Table 2 describes the basic functionality of each of the ARBAS-IOM code modules depicted in Figure 8.

C. PROCEDURES FOR ACCESSING AND RUNNING THE IOM DSS APPLICATION

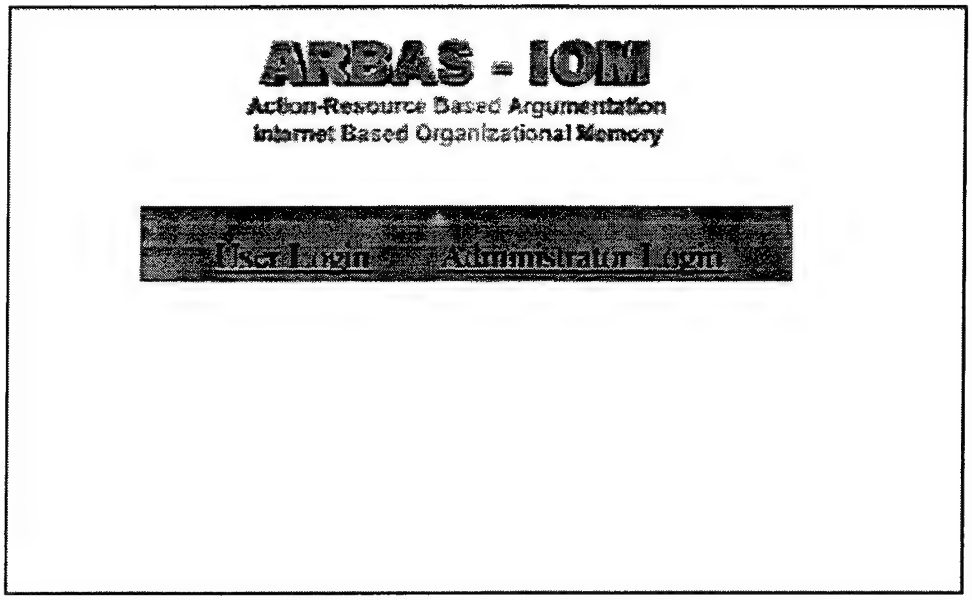
As discussed earlier, the IOM application uses the latest in Web based technologies. The application modules are stored in the root directory of the web server. Currently, the application resides at *http//131.120.41.223/Thesis/.* Users can access the application by utilizing any standard Web browser and the URL listed above. For the purpose of the prototype application, the user name '*Admin*' has been created with the password of '*password*'.

Table 2. ARBAS-IOM Module Descriptions

Name	Function
Index.cfm	Allows a user to select either User or Admin application functionality's
User.cfm	Provides the form for the user to specify last name and password
Authenticate.cfm	A script file that validates the users or denies access if incorrect login attempt
User_qorig.cfm	Displays all current on-line discussions; provides menus to create a new thread, complete a thread, view a thread, or query the database.
NT.cfm	Allows the user to initiate a new threaded negotiation or discussion. This module allows the user to specify the participants of the thread and the subject. Submits data to the database.
Newmess.cfm	Inserts the thread initiators information into the Thread table and obtains new information about the proposed problem, solution, and view. Submits data to database.
Thread2.cfm	Displays all of the messages that other users have sent, allows user to respond to these messages. Submits data to database.
Updatestatus.cfm	Allows user to complete threaded discussion and document final decision.
Updatestatus2.cfm	Submits the completed threaded discussion to the database.
Qom.cfm	Allows user to query database based on keywords or user last name
Results.cfm	Displays a concise list of results based on search criteria.
Detail.cfm	Displays a detailed list of results based on search criteria.
Admin.cfm	Provides the form for the Administrator to specify last name and password
Authenticate2.cfm	A script file that validates the Administrator or denies access if incorrect login
Admin_main.cfm	Displays all current users, provides menu to add or edit user profiles
Displayuser.cfm	Allows the Administrator to modify the users database information
New.cfm	Allows the Administrator to create a new user in the IOM application
Insertuser.cfm	Inserts the new user into the database.

Once the user has reached the indicated URL, they can choose whether to login as a normal user or login and perform administrator functions. The following section contains screen shots and explanations for each of the applications coded modules. In these particular screen shots the user, wright, goes through the process of adding a user, creating a new negotiation, responding to a discussion, completing a negotiation, and querying the database for information.

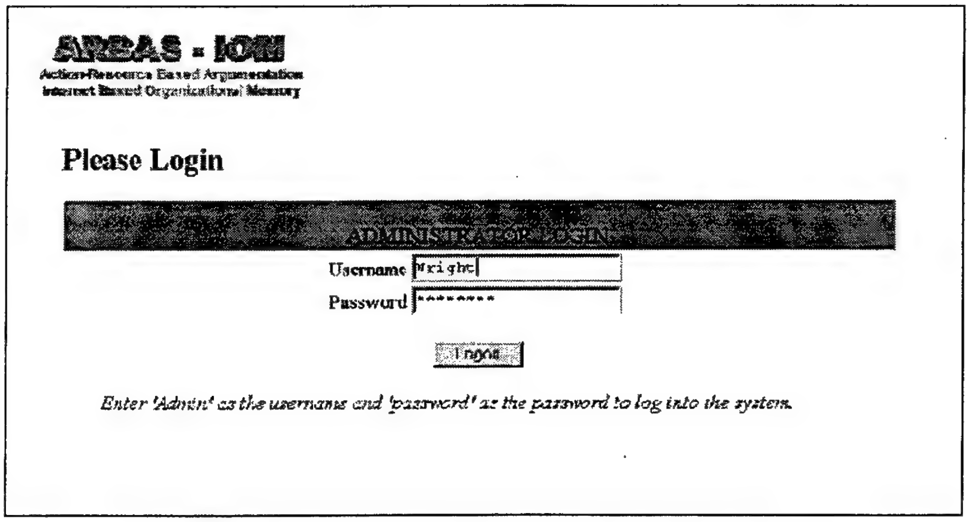
Figure 9 is the main screen for the ARBAS-IOM application. This screen allows the user to either login into the system as an administrator or regular user. The Administrator login allows the user to add new users or modify existing users in the database. The User login allows the user utilize the on-line negotiation and argumentation software.


ARBAS - IOM

Action-Resource Based Argumentation
Internet Based Organizational Memory

Figure 9. Main Application Menu Screen (Index.cfm)

From this screen (Figure 10), the user enters their user name and password, then selecting the "Logon" button. The module calls the Authentication.cfm, which will ensure that the user is authorized to perform administrative functions. If the user either incorrectly performs the login or is an unauthorized user, the system will display and error message. The user will be return to this screen with the opportunity to retry the login procedure as depicted in Figure 10.


ARBAS - IOM

Action-Resource Based Argumentation
Internet Based Organizational Memory

Please Login

ADMINISTRATOR LOGIN

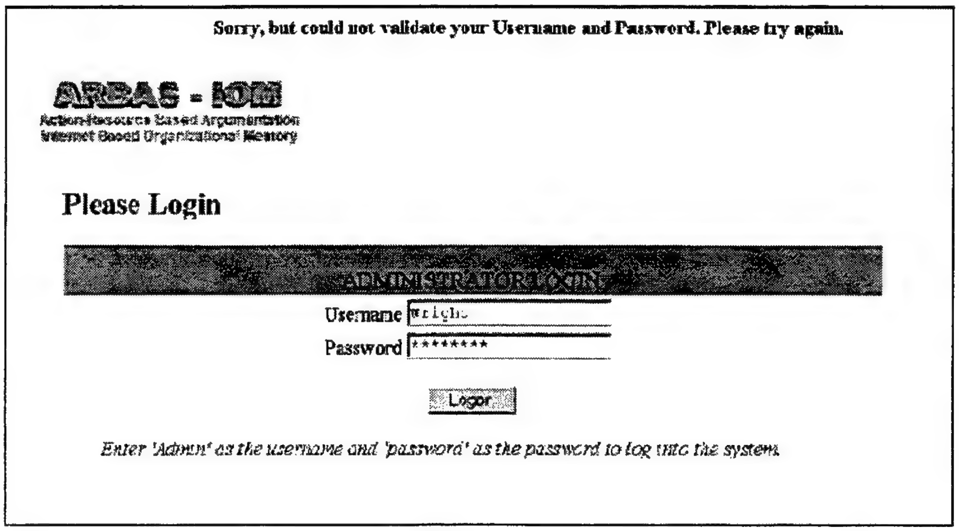
Username Wright

Password ********

Enter 'Admin' as the username and 'password' as the password to log into the system.

Figure 10. Administrator Login Screen (Admin.cfm)

The user was not validated in Figure 11 and must try and login to the application again.



Sorry, but could not validate your Username and Password. Please try again.

ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory

Please Login

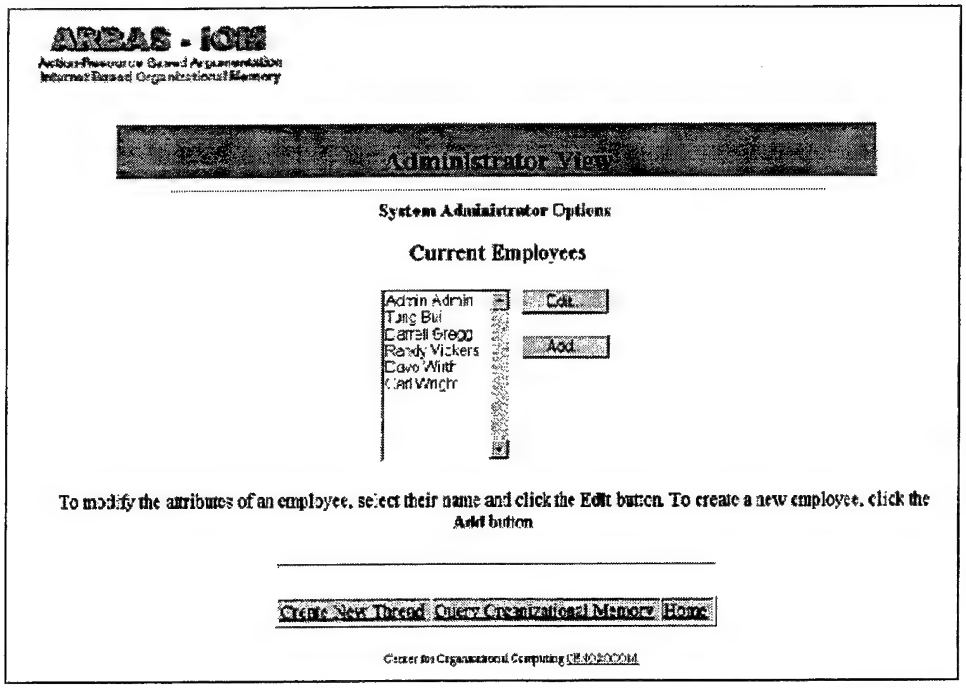
Username: wright
Password: ********

Enter 'Admin' as the username and 'password' as the password to log into the system.

Figure 11. Administrator Login Screen (Admin.cfm)

From the main system administrator screen (Figure 12), the user can select to edit or add new users into the database. The users that appear in the menu box are queried from the database and inserted into the form field. This is one of the many dynamically generated web-centric fields, which significantly reduce the maintenance of web pages as they related to changes that happen in a more or less consistent manner. Dynamically generated information also insures that the administrator is always looking at the most current list of users.

The Displayuser module executes two functions. The first action is to query the database based on the user selected in the Admin_main module and insert the results of that query into the form seen in Figure 13. The second action is to re-insert the updated user's information into the database. This process is not as simple as it looks. The application is required to update the record in the database, not create a new instance. Once the Administrator has modified the users information and selects "Update Record", the data is inserted into the database.


ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory
Administrator View
Edit...
Add...
Create New Thread
Query Organizational Memory
Home
Center for Organizational Computing

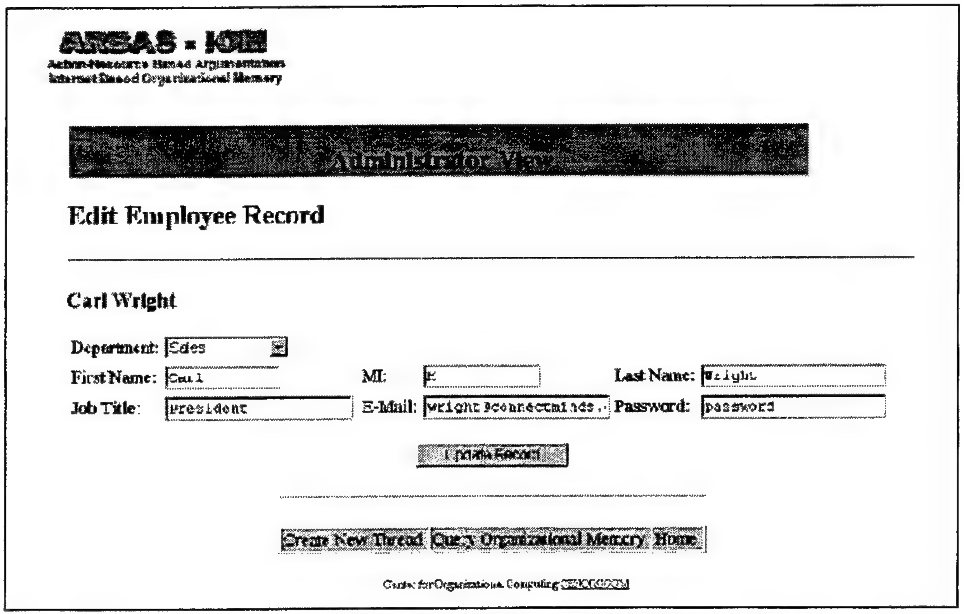
System Administrator Options

Current Employees

Admin Admin
Tang Bui
Carrell Gregg
Randy Vickers
Dave Wirth
Carl Wright

To modify the attributes of an employee, select their name and click the **Edit** button. To create a new employee, click the **Add** button

Figure 12. Administrator Main Screen (Admin_main.cfm)


ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory
Administrator View
Update Record
Create New Thread
Query Organizational Memory
Home
Center for Organizational Computing

Edit Employee Record

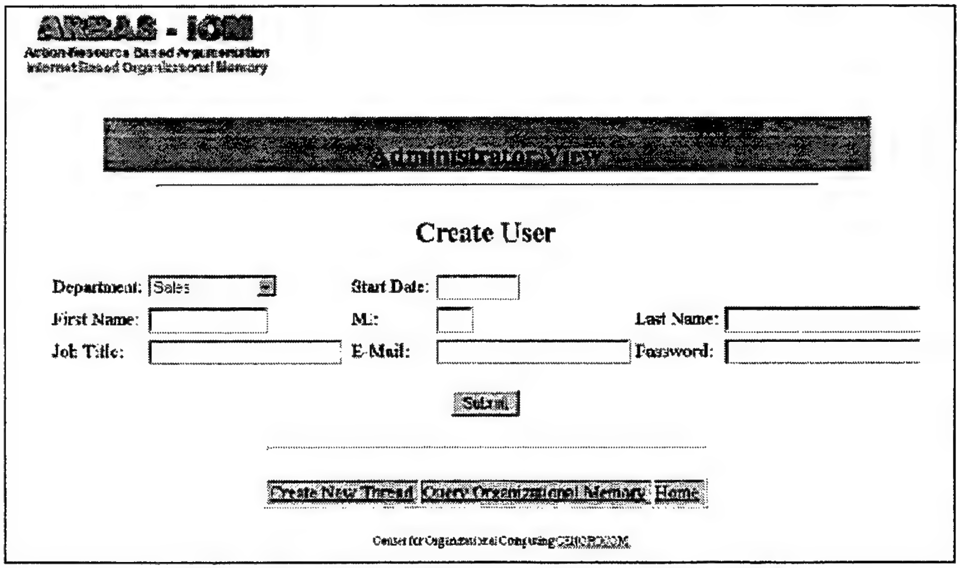
Carl Wright

Department:	Sales				
First Name:	Carl	MI:	E	Last Name:	Wright
Job Title:	president	E-Mail:	wright@connectmindS..	Password:	password

Figure 13. Edit User Screen (Displayuser.cfm)

This module (Figure 14) application allows the Administrator to create and insert new users into the database. Once the Administrator has "Submitted" the information, a new instance of the user is created in the database. From this point on, the user will be able to participate as an active member in negotiation and discussion.

Similar to the Administrators login module (Figure 15), the User login module also validates the users name and password. If the user is an authorized, then they will be allowed to enter the main user menu for the IOM application.


ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory
Administrator View
Submit
Create New Thread
Query Organizational Memory
Home

Create User

Department: Sales

Start Date:

First Name:

MI:

Last Name:

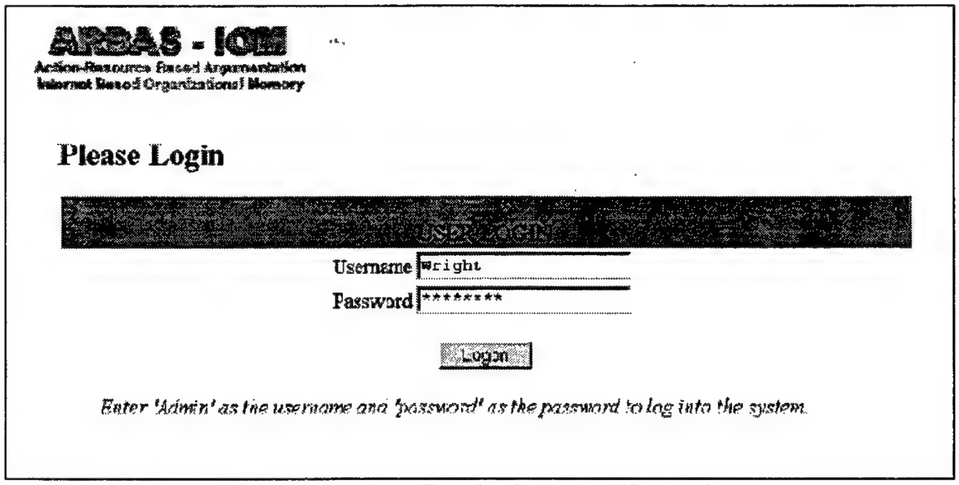
Job Title:

E-Mail:

Password:

Figure 14. Create New User Screen (New.cfm)

The main user menu, Figure 16 is a screen-shot demonstrating a sample user screen.


ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory
USER LOGIN
Logon

Please Login

Username wright

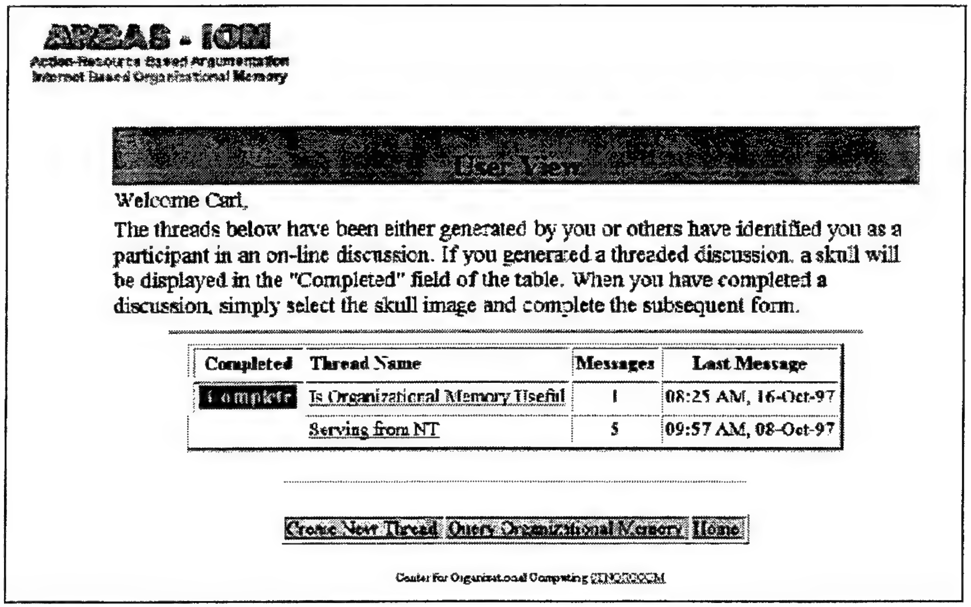
Password ********

Enter 'Admin' as the username and 'password' as the password to log into the system.

Figure 15. User Login Screen (User.cfm)

From this menu the user can execute various functions within the IOM application. The *User_qorig* module personalizes the application for each user based on login name. Each user is welcomed with their first name. The module also queries the database, searching for any open negotiations or discussions that the user might be involved in. A user can be designated to participate in a thread in two ways. The first way, the user was identified by another user as an active participant in a discussion, as seen in Figure 16, "Serving from NT". Secondly, a user can initiate a threaded discussion, such as, "Is Organizational Memory Useful". In this case, the user created the thread, so a "Complete" tag is displayed next to the thread name. If the user were to "click" on the image, they would be allowed to complete a form that would end the threaded negotiation/discussion. The module checks the database, only displaying the threads for this user.

If the user selects one of the threads under the caption "Thread Name", they will be taken to the module which displays the messages from others users that are concurrently participating in the discussion. The user is also given the opportunity to respond to any of the messages in the thread. All responses are stored in the database.


ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory
User View

Welcome Carl,

The threads below have been either generated by you or others have identified you as a participant in an on-line discussion. If you generated a threaded discussion, a skull will be displayed in the "Completed" field of the table. When you have completed a discussion, simply select the skull image and complete the subsequent form.

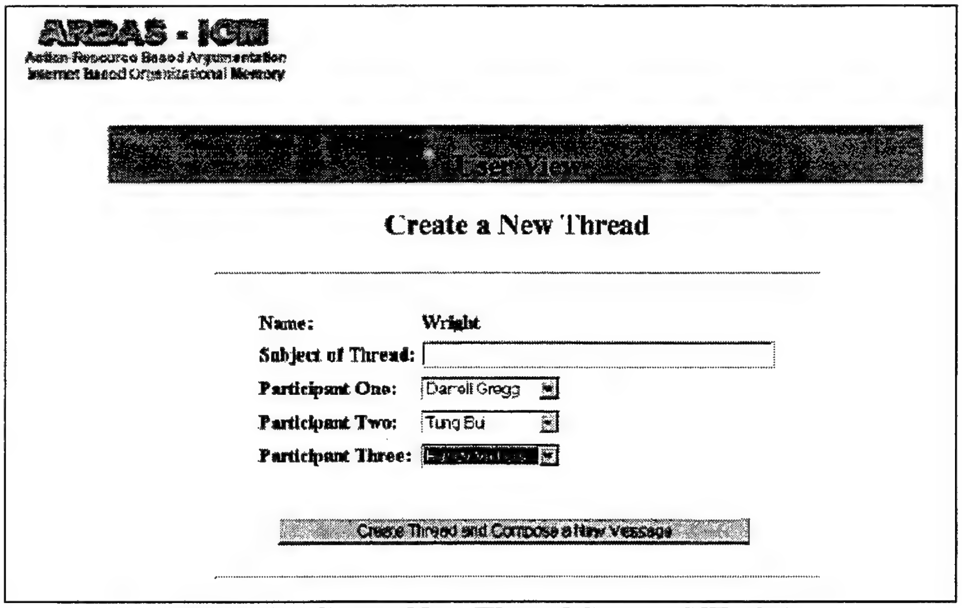
Completed	Thread Name	Messages	Last Message
Complete	Is Organizational Memory Useful	1	08:25 AM, 16-Oct-97
	Serving from NT	5	09:57 AM, 08-Oct-97

Create New Thread
Query Organizational Memory
Home
Center For Organizational Computing [illegible]

Figure 16. User Main Screen (User_qorig.cfm)

At the navigational bar, the user is allowed to create new threaded discussions, query the database for information, or go back to the main menu screen, Figure 9.

Figure 17 is a screen shot of the Create New Thread module. The thread initiator's name (the user), is stored as a Cookie throughout the session of the application. Thus, the user is not required to re-enter their personal information for every transaction. In this module the user/initiator is allowed to indicate the subject of the thread and the participants. The participant selection boxes are automatically updated from the database so that they contain the most current users available for on-line discussions. Once the user has filled out the mandatory fields, the information is inserted into the database, creating a new instance of a thread. This will also update the other users profiles, insuring that when they login to the system they are included in the threaded discussion.


ARBAS - IOM
Action Resource Based Argumentation
Internet Based Organizational Memory
User View

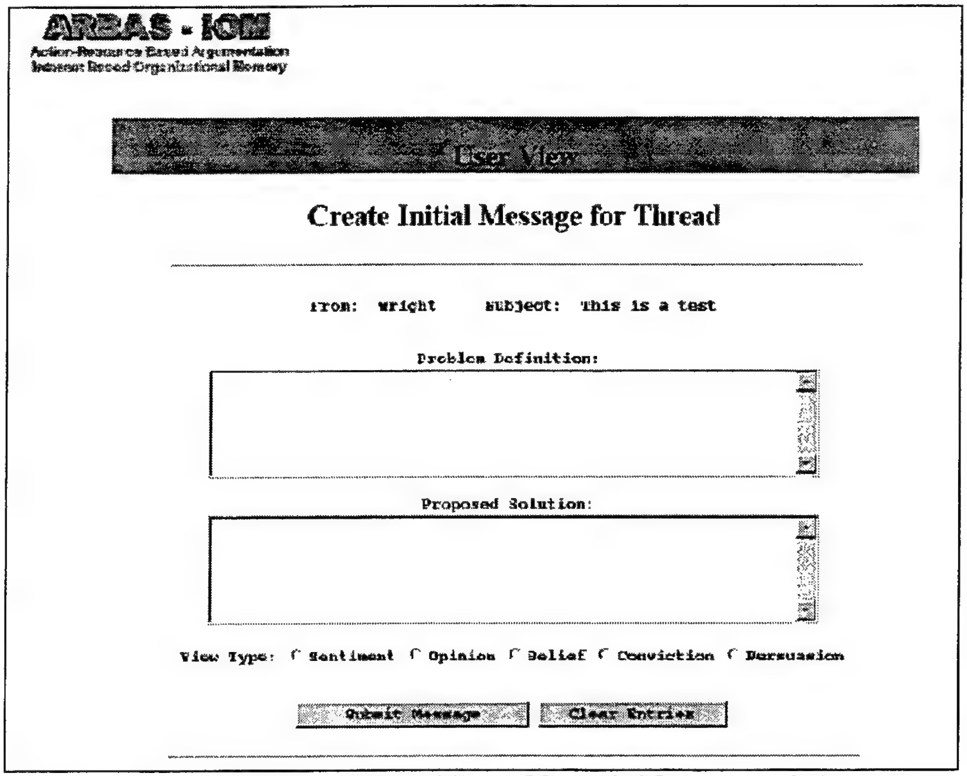
Create a New Thread

Name: Wright
Subject of Thread:
Participant One: Darell Gregg
Participant Two: Tung Bui
Participant Three: [illegible]

Create Thread and Compose a New Message

Figure 17. Create New Thread Screen (NT.cfm)

The screen in Figure 18 allows the user to define the problem and suggest a proposed solution. The user also identifies their view type. This information is submitted into the database and is related to the thread.



ARBAS - IOM

Action-Resource Based Argumentation Internet Based Organizational Memory

User View

Create Initial Message for Thread

From: wright   Subject: This is a test

Problem Definition:

Proposed Solution:

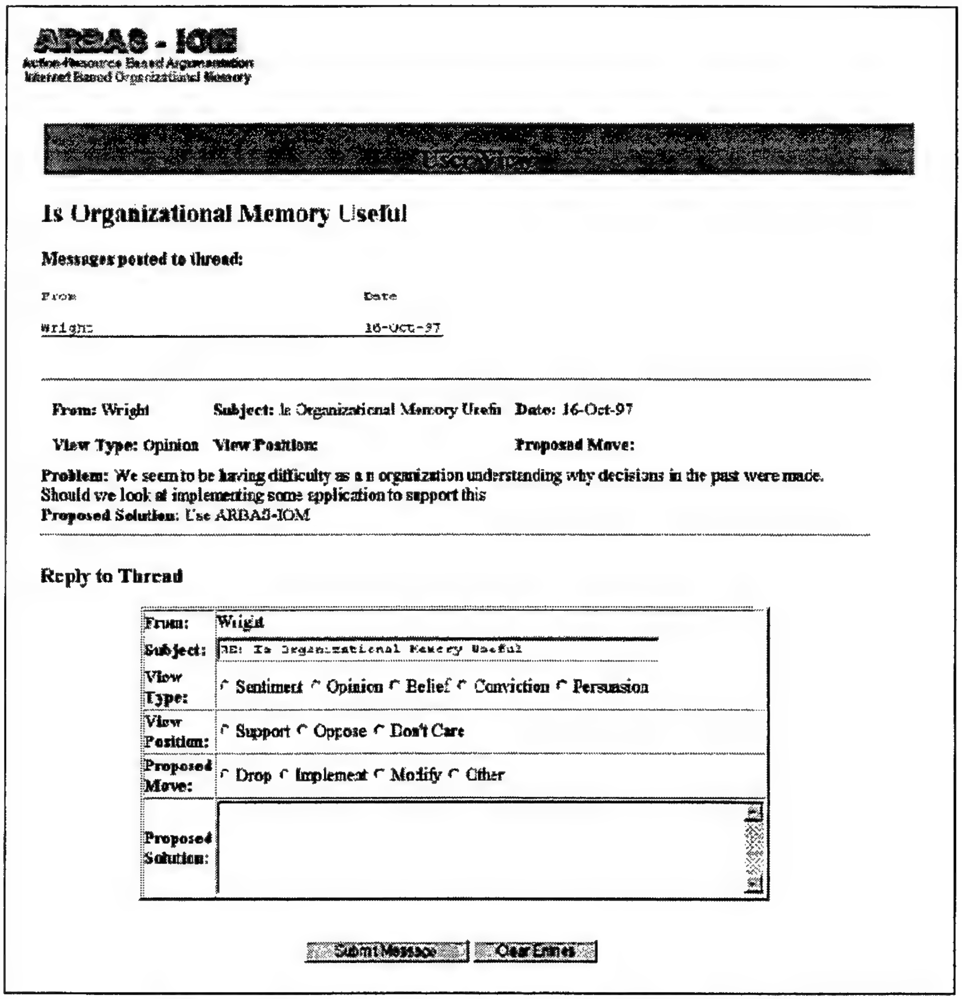
View Type: ○ Sentiment ○ Opinion ○ Belief ○ Conviction ○ Persuasion

Figure 18. Create New Thread Part 2 (Newmess.cfm)

The screen in Figure 19 displays the messages associated with the threaded negotiation/discussion. Users are allowed to view all of the information other users have input and able to make informed responses. Any information in the "Reply to Thread" form is inserted into the database and associated with the initial thread.

The screen in Figure 20 allows the initiator of a threaded discussion to close the thread. The user is required to indicate what the final decision is and the justification behind that decision. This information is inserted into the database. This module will also update the other user's profiles so that this instance of a threaded discussion will not appear on their main user menu.

ARBAS-IOM offers robust Query capability. The module in Figure 21 provides the user the ability to query historical threads based on keywords, user last names, or a combination of both. If the fields are left blank, the application will return every instance of a threaded discussion in the database.


ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory
User Views

Is Organizational Memory Useful

Messages posted to thread:

From	Date
wright	16-Oct-97

From: Wright Subject: Is Organizational Memory Usefu Date: 16-Oct-97

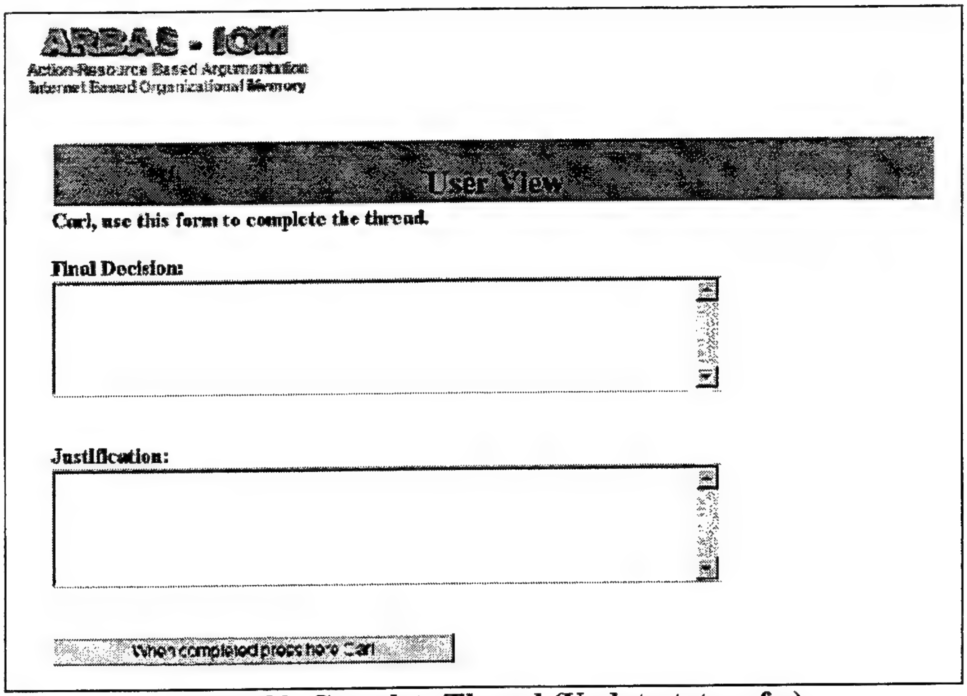
View Type: Opinion View Position: Proposed Move:

Problem: We seem to be having difficulty as a n organization understanding why decisions in the past were made. Should we look at implementing some application to support this
Proposed Solution: Use ARBAS-IOM

Reply to Thread

From:	Wright
Subject:	RE: Is Organizational Memory Useful
View Type:	○ Sentiment ○ Opinion ○ Belief ○ Conviction ○ Persuasion
View Position:	○ Support ○ Oppose ○ Don't Care
Proposed Move:	○ Drop ○ Implement ○ Modify ○ Other
Proposed Solution:	

Figure 19. View/Respond Thread (Thread2.cfm)


ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory
User View

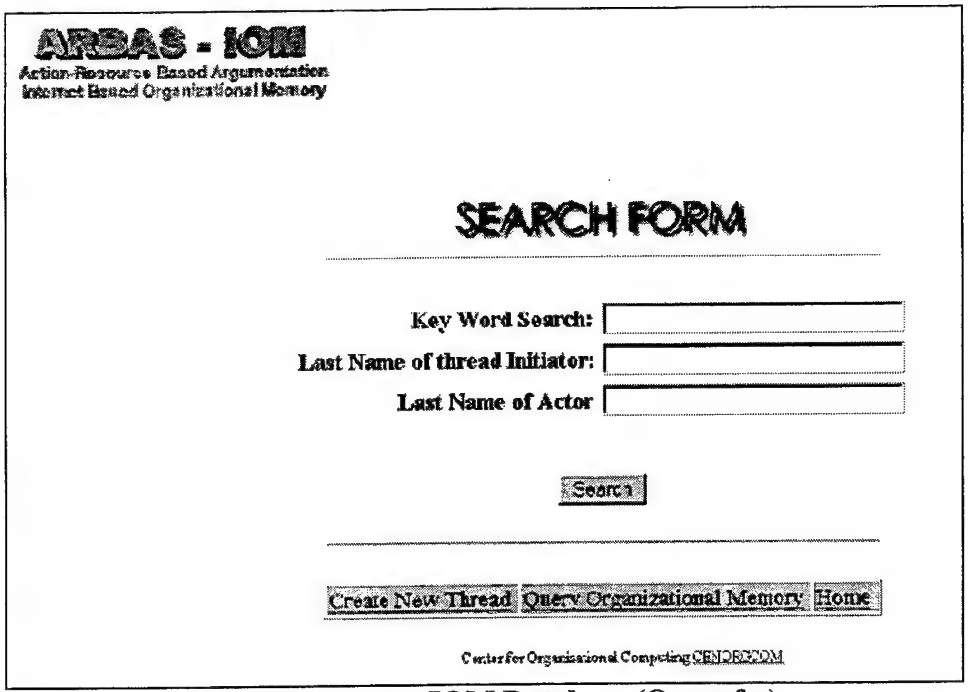
Carl, use this form to complete the thread.

Final Decision:

Justification:

When completed press here Carl

Figure 20. Complete Thread (Updatestatus.cfm)


ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory
SEARCH FORM

Key Word Search:

Last Name of thread Initiator:

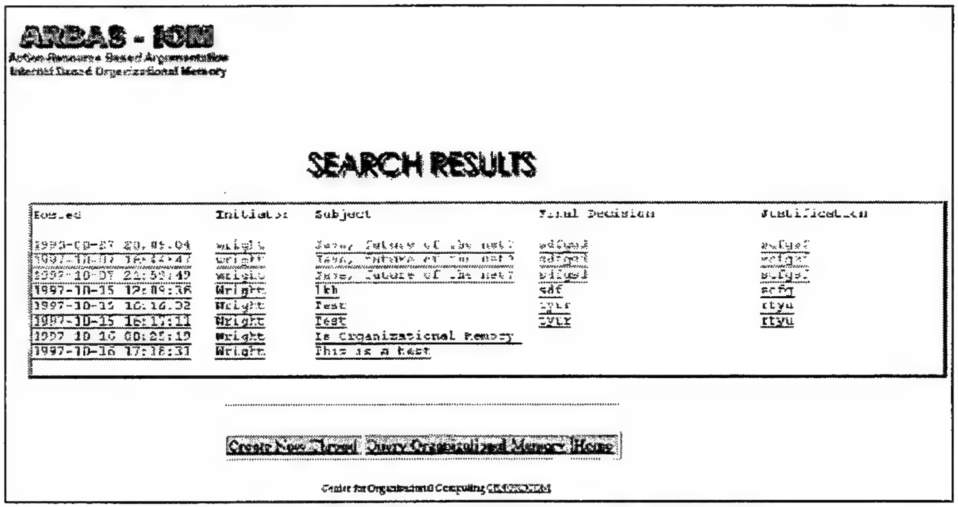
Last Name of Actor

Search
Create New Thread
Query Organizational Memory
Home
Center for Organizational Computing CENORGCOM

Figure 21. Query IOM Database (Qom.cfm)

The results of the user query are displayed in a table as seen in Figure 22. The search returns each instance of a record that matches the users search criteria. Each of the records above are linked to a detailed web page that will provide all the information in the database with regard to that instance.

The result of the users detailed query is also displayed in a table format (Figure 23).

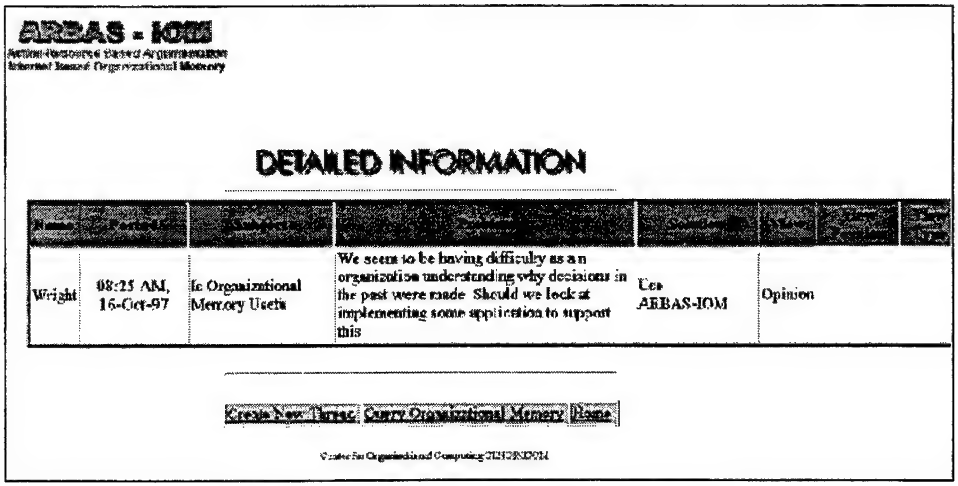

ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory
Create New Thread
Query Organizational Memory
Home
Center for Organizational Computing

SEARCH RESULTS

Posted	Initiator	Subject	Final Decision	Justification
[illegible] 20:05:04	wright	Java, future of the net?	[illegible]	[illegible]
[illegible]	wright	Java, future of the net?	[illegible]	[illegible]
1997-10-07 21:52:49	wright	Java, future of the net?	[illegible]	[illegible]
1997-10-15 12:09:36	Wright	lkh	sdf	scfg
1997-10-15 16:16:32	Wright	Test	tytr	rtyu
1997-10-15 16:17:11	Wright	Test	tytr	rtyu
1997-10-16 08:25:19	Wright	Is Organizational Memory		
1997-10-16 17:18:31	Wright	This is a test		

Figure 22. Results of the Query (Results.cfm)

The user can now see all the detailed information of the records associated with the criteria of their search.


ARBAS - IOM
Action-Resource Based Argumentation
Internet Based Organizational Memory
Create New Thread
Query Organizational Memory
Home
Center for Organizational Computing

DETAILED INFORMATION

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Wright	08:25 AM, 16-Oct-97	Is Organizational Memory Useful	We seem to be having difficulty as an organization understanding why decisions in the past were made. Should we look at implementing some application to support this	Use ARBAS-IOM	Opinion		

Figure 23. Detailed Query (detail.cfm)

V. SUMMARY AND RECOMMENDATIONS FOR FUTURE RESEARCH

A. SUMMARY

The purpose of this thesis was to explore the feasibility and development of using various web centric technologies to provide access to corporate knowledge which enhances the decision making process. Current trends in web and network technologies allow applications like ARBAS-IOM to be developed and effectively utilized in the decision making process. This research supports the idea of using the Internet, an Intranet, or an Extranet as a viable means to provide a distributed medium to create and store organizational memory. The only requirements for the organization to use this application are networked computers with access to a web browser like Netscape Navigator.

The only limiting factor for this application is storage space on the computer that holds the database. As problems are realized and threads are generated the database will fill rapidly. A possible remedy would be limiting the length of the threads or the number of responses per problem. Another possibility would establish a dedicated server and storage device for the application. As long as the network does not become too crowded, bandwidth will not be an issue. There are no graphics in the current application therefore bandwidth use is maximized.

The basic systems development life cycle, SDLC, was not used during the development of the ARBAS-IOM application. Instead of the six basic steps of the SDLC (Problem recognition/definition, Feasibility study, Analysis, Design, Implementation, Testing, and Maintenance), prototyping was used. Prototyping has four basics steps in the development of an application. The four steps are:

- Identify the user's basic requirements,
- Develop an initial prototype,
- Use the prototype,

- Revise and enhance the prototype.

The development cycle of ARBAS-IOM deviated slightly in the sense that multiple prototypes were not developed. Changes were made in the application to improve the interface, but the functionality was not changed. As is, the application can be implemented in an organization wishing to track their decision making process. The following list provides an estimate of the percentage of time spent in each phase of the prototyping method.

• Identification of user's requirements	10%
• Development of initial prototype	60%
• Use of the prototype	20%
• Prototype revision	10%

B. LESSONS LEARNED

The tools available today used to create web based negotiation systems provide solid and easy to use resources. As inexperienced developers of this type of applications, the software used to build ARBAS-IOM were relatively intuitive and well supported by the tool's documentation and on-line support. There were also a number of good "How-to" books available by professional users and developers of then tools. Some prior knowledge of browser technologies, database development, and programming the Cold Fusion metafiles helped to speed up the development, but was not required. We recommend a computer with at least 32 MB RAM, a state of the art CPU, and enough storage space to hold the development software, the database, and the application help decrease development time also.

The greatest challenge to development process was learning detailed SQL and middleware coding. Learning the Cold Fusion language took some time away from the development process, but as with any language continued use increases proficiency. Learning the technology that the application is being built with prior to the beginning of the project will decrease the actual development time significantly. The prior knowledge mentioned above helped but a steep learning curve was still evident.

C. RECOMMENDATIONS FOR FUTURE RESEARCH

Before future development, alternate design tools should be considered. As technology changes and software becomes more "point-and-click" the development of such an application can be made easier. Web site development tools are becoming more integrated with database interfaces as the need to store and retrieve data increases in demand.

Storage media needs consideration since the database associated with such an application can become extremely large. Optical media, DAT tape devices, and other high performance storage devices can be considered during the development process.

Future development and integration with other essential business process and applications is the key to the enhancements of applications that enhance an organizations decision making process.

LIST OF REFERENCES

Ackerman, Mark S., "Organizational Memory," URL: http://www.ics.uci.edu/~ackerman/om.html

Allaire Corporation, *Cold Fusion 3.0,* July 1997.

Axelrod, R., "Argumentation in Foreign Policy Settings: Britain in 1918, Munich in 938, and Japan 1970," in *The Negotiation Process*, (Zartman, Ed.), Sage, Beverly Hills, CA., 1977

Ballmer, T. and Brennenstuhl, W., Speech Act Classification: A Study in the Lexical Analysis of English Speech Activity Verbs, Springer-Verlag, Berlin. 1981.

Binbasioglu, M. and Jarke, M, "Process Based Reconstructive Approach to Model Building," *Decision Support Systems*, 1986.

Bui, T., "Evaluating Negotiation Support Systems: A Conceptualization," in *Proceedings of the 27th Hawaii International Conference on System Science*, Maui, 1994.

Bui, T. and Shakun, M.F., "Negotiation Processes, Evolutionary systems Design and NEGOTIATOR," *Group Decision and Negotiation*, 1995.

Chang, A-M and Han T-D., "Design of an Argumentation-Based Negotiation support System," in *Proceedings of the 28th Hawaii International Conference on System Science*, Maui, 1995.

Chang, M-K and Woo C., "A Speech-Act-Based Negotiation Protocol: Design, Implementation, and Test Use," *ACM Transactions on Information Systems*, 1994.

Davis, R. and Smith R.G., "Negotiation as a Metaphor for Distributed Problem Solving," *Artificial Intelligence*, 1983.

Ehrlich, Kate, "Issues In Retrieving Information From Organizational Memory" URL: http://www.mcs.net/~pgustafs/Latest_Prototype/pres-ehrlich-9-94.html.

Fang, L., Hipel and Kilgour, *Interactive Decision Making, The Graph Model for Conflict Resolution,* ,New York, 1993.

Forta, Ben, *the Cold Fusion: Web Database Construction Kit*, Que Corporation, Indianapolis, In, 1997.

Hackman and Morris, "Group Tasks, Group Interaction Process, and Group Performance Effectiveness: A Review and Proposed Integration," in L. Berkowitz (ed.), *Advances in Experimental Social Psychology*, Vol. 8, Academic Press, New York, 1975.

Haramundanis, K., *The Art of Technical Documentation*, Digital Press, 1992.

Hegel, G. W. F., *The Phenomenology of Mind*, Second Edition, Translated by J. B. Baillie, George Allen and Unwin, Ltd., London, 1964.

Hiltz S.R. and Turoff M., *The Network Nation: Human Communication via Computer*, Addison-Westley, New York, 1978.

Keough, C.M. and Lake, R.A., "Value as Structuring Properties of Contract Negotiations," in Values, Communications, and Organizational Decisions, Conrad (Ed.), Ablex, Norwood, NJ. , 1991.

Kersten, G., "Simulation and Analysis of Negotiation Processes: The Case of Softwood Lumber Negotiations," in *Proceedings of the 28th Hawaii International Conference on System Science*, Maui, 1995.

Kersten, G. and Cray, D., "Perspectives on Representation and Analysis of Negotiation" URL: http://interneg.carleton.ca/interneg/research/misc/boston_gdn/.

Kersten, G. and Noronha, S., "Negotiation Via the World Wide Web: A Cross-Cultural Study of Decision Making" URL:http://interneg.carleton.ca/interneg/research/misc/inc9.html.

Koo, C.C., "Office Are Open Systems," ACM Transactions on Office Information Systems, Vol. 4, No. 3, 1988.

Lee, J., "SIBYL: A Tool for Managing Group Decision Rationales" *in Proceeding of Conference on Computer-Supported Cooperative Work*, ACM Press, Los Angeles, 1990.

Lotus Development Corp., *Lotus Notes; Release 3.0 c*, 2. (December 1993).

McAllester, D., "Reasoning Utility Package User's Manual, Artificial Intelligence Laboratory," *AI Memo 667*, MIT, Cambridge, MA 1982.

Melard, P and Vanreusel, J.F., "Memoire organisationelle et systeme d'aide a la decision: ARBAS'96," Memoire de diplome, Facultes Universitaires Notre-Dame de la Paix, Namur, 1996.

NetObjects, INC., *NetObjects Fusion 2.01*, January 1997.

Ramesh, R. and Whinston A.B., "Claims, Arguments, and Decisions: Formalisms for Representation, Gaming, and Coordination," *Information Systems Research,* September 1994.

Smith, R.G., and Davis R.,"Framework of Cooperation in Distributed Problem Solving," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 11, No. 1., 1981.

Sawyer, J. and Guetzkow, H., "Bargaining and Negotiation in International Relations," *International Behavior: A Social-Psychological Analysis*, Holt, Rinehart and Winston, New York, 1965.

Thomas, K.W., "Conflict and Negotiation Processes in Organizations:, in M.D. Dunnette (ed.), *Handbook of Industrial and Organizational Psychology*, 2nd Ed., Vol, 3, Consulting Psychologists Press, Palo Alto, 1992.

Toulmin, S.E., *The Use of Arguments*, Cambridge University Press, Cambridge, Massachusetts, 1958.

Toulmin, S. E., Rieke R. and Janik A., *An Introduction to Reasoning*, Macmillan, New York, 1979.

O'Reilly Software, O'Reilly & Associates, Inc., *WebSite Professional 1.0*, September 1997.

Zartman, I.W. and Berman, M., *The Practical Negotiator*, Yale University Press, New Haven, 1982.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
8725 John J. Kingman Road, Suite 0944
Fort Belvoir, VA 22060-6218

2. Dudley Knox Library 2
Naval Postgraduate School
411 Dyer Rd.
Monterey, CA 93943-5101

3. Prof. Tung X. Bui 1
The University of Hawaii
College of Business
2404 Maile Way
Honolulu, HI 96822

4. Prof. Suresh Sridhar 1
Naval Postgraduate School
Monterey, CA 93943-5103

5. Randal R. Vickers 2
2015 Crestwood Drive
Richmond, Tx 77469

6. Carl M. Wright 2
c/o Sally Bisbee
1220 River Terrace drive
Bloomington, Mn 55431

7. Director, Training and Education 1
MCCDC, Code C46
1019 Elliot Road
Quantico, VA 22134-5107

8. Director, Marine Corps Research Center 2
MCCDC, Code C40RC
2040 Broadway Street
Quantico, VA 22134-5107

9. Director, Studies and Analysis Division 1
MCCDC, Code C45
3300 Russell Road
Quantico, VA 22134-5130

10. Marine Corps Representative 1
Naval Postgraduate School
Code 37, Bldg. 234, HA-220
699 Dyer road
Monterey, CA 93940

11. Marine Corps Tactical Systems Support Activity.. 1
Technical Advisory Branch
Attn: Maj. J. C. Cumminsky
Box 555171, Camp Pendleton, CA 92055-5080